mc

März 1988
7 DM · 60 öS · 7 sFr

Die Mikrocomputer-Zeitschrift

3/88

Sensation zur CeBit:

Z-386 mit OS/2

für Wissenschaft und Technik

PC-Technik:
AD-Wandler für Professionals

Amiga:
Karikaturen zeichnen

Atari ST:
Devpac-Assembler

C-Compiler:
Neues Quick-C für PC
Großer Vergleichstest:
Die besten C-Compiler für den Atari ST

Was Sie über Modula wissen sollten

PLANTRON

Hier ist der neue STANDARD!

Das Raumwunder der AT-Generation!

PT-286 AT
mit 64 MB Festplatte und Super-EGA-Karte:

* 80286 CPU - 6/10 MHz schaltbar
* 640 KB RAM (bis 1 MB on Board)
* Sechs 16-bit und zwei 8-bit Slots
* CMOS-Speicher mit Uhr/Kalender
* Parallele und serielle Schnittstelle
* Festplatten-/Diskettencontroller
* 1.2 MB Diskettenlaufwerk
* 64 MB (netto) Festplatte
* Super-EGA-Karte bis 640 x 480 Punkte oder 132 Spalten bis 1056 x 352 Punkte
* Große DIN-Tastatur
* Erweitertes MS-DOS 3.20 und GW-BASIC

DM 4998,–

AUTOSCAN /2
Super-EGA-Monitor
DM 1698,–

32-bit High-Tech zum AT-Preis!

PT-386 HT / 2
mit 32 MB Festplatte und Monochrom-Grafikkarte:

* Sonstige Ausstattung wie PT-386 HT

DM 7998,–

PT-386 HT
mit 64 MB Festplatte und Super-EGA-Karte:

* 80386 CPU - 16 MHz
* 512 KB 32-bit RAM
* Zwei 32-bit, vier 16-bit und zwei 8-bit Slots
* CMOS-Speicher mit Uhr/Kalender
* Zwei parallele und zwei serielle Schnittstellen
* Festplatten-/Diskettencontroller
* 1.2 MB Diskettenlaufwerk
* 64 MB (netto) Festplatte
* Super-EGA-Karte bis 640 x 480 Punkte oder 132 Spalten bis 1056 x 352 Punkte
* Große DIN-Tastatur
* Erweitertes MS-DOS 3.20 und GW-BASIC

DM 9998,–

Testbericht in CHIP 7/'87

PT-386 HT (Level 2) jetzt mit folgender Standardausstattung:
– 1 MB RAM (bis 2 MB on Board)
– Sockel für 80287- und 80387-Coprozessoren
– Verlagerung des ROM-BIOS und des EGA-ROM in das 32-Bit-Extended-RAM (Geschwindigkeitsverbesserung bis 300 % [BIOS] bzw. 1200 % [EGA])

Alle EGA-Systeme jetzt mit neuer Super-EGA-Karte (Level 5):
– 800 x 600 Punkte Auflösung
– Erweiterte Steuersoftware

Alle Computersysteme werden mit ausführlichem deutschen Handbuch geliefert.
PLANTRON-Produkte erhalten Sie nur im autorisierten Fachhandel.
Bitte fordern Sie Unterlagen zu unserem Gesamtlieferprogramm sowie das Fachhändlerverzeichnis an.

Alle Preisangaben sind unverbindliche Preisempfehlungen.

PLANTRON Computer GmbH · D-6380 Bad Homburg · Telefon: 06172/25188* · Telex: 417410 placo d · Telefax: 06172/81033

EDITORIAL

Liebe Leser!

In diesem Heft geht es um einen starken Rechner: den Z-386 von Zenith. Schon die Hardware allein wäre stark, zusammen mit der zugehörigen Software entsteht eine kaum noch zu schlagende Kombination für den „gehobenen Computeristen". Besonders geeignet ist das Gerät für den technisch-wissenschaftlichen Bereich, in dem es seine Stärken voll ausspielen kann. Unser Spezialist Dieter Strauß hat für Sie ausprobiert, wie sich die OS/2-Implementation von Zenith auf dem Gerät anfühlt. Manches ist ja erst richtig erkennbar, wenn man es in der Praxis betrachtet. Wie schnell ist OS/2 wirklich, wie komfortabel die Benutzeroberfläche für den Anwender und den Systemspezialisten? Der Test des Z-386, eines richtigen 32-Bit-Rechners (kein 16-Bit-Bus!) hat es in sich.

Kennen Sie Ray Duncan? Vielleicht haben Sie schon einmal den Namen gehört? Wer sich mit PCs beschäftigt, der hat mit Sicherheit schon von ihm gehört und von ihm profitiert. Er ist d e r MS-DOS- und PC-Kenner in den USA. Ray Duncan wird für mc aus den USA berichten. In lockerer Folge wird er die Szene in den USA beleuchten und Trends und Tendenzen schildern, wie er sie sieht. Das ist wichtig für uns, denn das Geschehen in den USA beeinflußt den professionellen Markt in der Bundesrepublik. Bewußt habe ich „professionell" gesagt, denn die Entwicklung des Heimcomputermarktes bei uns verläuft unabhängig von den USA. Commodore ist hier eine große Nummer, in den USA zwar nicht völlig unbekannt, aber eben eine Firma unter vielen. Atari ist in der Bundesrepublik stärker im Markt und erfolgreicher als in den USA. Was Ray Duncan zum Beispiel zu OS/2 und dem PC-Geschehen zu sagen hat (nicht unkritisch!), das können Sie in dieser mc-Ausgabe lesen.

Zwei Mitarbeiter in der Abteilung Systementwicklung und Schulung des WDR haben ein Projekt entwickelt, das Videotext jetzt erst wirklich nutzbar macht. Videotext, das ist der Informationsdienst, der mit dem Fernsehbild in Zeilensignalen übertragen wird, die im Normal-Fernsehgerät wohl empfangen, aber nicht auf dem Bildschirm dargestellt werden. In dem Beitrag „Videotext in den Computer" geht es um die Überleitung der gebotenen Informationen in einen (Standard-)Computer, um sie dort weiter bearbeiten zu können. Die Schaltung wartet mit einigen technischen Leckerbissen auf und ergänzt so nebenbei Fernsehgeräte um einen Videotext-Decoder. In diesem Heft gibt es die Grundlagen und einen Überblick über die Funktionen. Die Platine wird im Handel erhältlich sein. Wolfgang Back wird das Ganze im Rahmen seiner Sendung „Kaleidoskop" (WDR, drittes Programm) in der „Computer-Club-Ekke" vorstellen. Termin: Samstag, 5. März, 17 bis 18 Uhr.

Prolog ist eine faszinierende Programmiersprache – wenn man sie versteht. mc hat den Beweismechanismus von Prolog untersucht. Aus technischen Gründen mußte in diesem Beitrag das Literaturverzeichnis weggeschnitten werden. Hier ist es, Lesefutter für abstrakte Denker: Böhme, G.: Einstieg in die Mathematische Logik. Hanser, München, 1981. Clocksin, W. F./Mellish, C. S.: Programming in Prolog. Springer, Berlin, 1984. Fröschl, K. A.: Logik als Programmiersprache, 1. Teil. In Österreichische Zeitschrift für Statistik und Informatik, Heft 2-3/1985, S. 179–215. Fröschl, K. A.: Logik als Programmiersprache, 2. Teil. In Österreichische Zeitschrift für Statistik und Informatik, 4/1985, S. 277 bis 295. Rich, E.: Artificial Intelligence. McGraw-Hill, New York, 1983. Robinson, J. A.: A Machine-Oriented Logic based on the Resolution Principle. In JACM, 12/1965, S. 23–41. Stede, M.: Einführung in die Künstliche Intelligenz. Hannover, 1987. Bleibt noch viel Freude an mc zu wünschen,

Ihr

Ulrich Rohde

DIE
UNIC-
ICB-IDEE

XT/AT-KOMPATIBLE COMPUTER IN INDUSTRIELLEM DESIGN

Die starke Verbreitung von IBM-kompatiblen Rechnersystemen hat es mit sich gebracht, daß eine sehr große Anzahl von entsprechenden Betriebssystemen, Programmiersprachen, Utilities und Programmgeneratoren rasch und preisgünstig erhältlich ist.

Die Verwendung dieser Programme für den industriellen Einsatz (Maschinensteuerung, Betriebsdatenerfassung) scheiterte meist daran, daß normale Rechnersysteme für rauhe Umgebungsbedingungen nicht ausgelegt waren (erschütterungsempfindlich, Probleme mit Staub, Temperatur und Feuchtigkeit).

VORTEILE DER UNIC-ICB-RECHNER

100 % softwarekompatibel zu IBM XT/AT-Rechnern. Vollständiger Aufbau auf Doppel- und Einfach-Europakarten – genormte Gehäuse, beliebige Schutzarten. Erweiterter Temperaturbereich durch entsprechende Steckverbindungen und Bauteileauswahl. Erschütterungsunempfindlich durch Einsatz nichtrotierender Massenspeicher (RAM/EPROM-Floppy).

Vollständig modular aufgebautes System, dessen Komponenten für Softwareentwicklung und eventuell Debugging auch am Personalcomputer angeschlossen werden können.

Original IBM-Erweiterungskarten (Netzwerkkarten, verschiedene Schnittstellenkarten usw.) sind über eine Adapterkarte direkt auf ICB-Systemen einsetzbar.

HANNOVER MESSE CeBIT'88 Welt-Centrum Büro-Information-Telekommunikation 16. – 23. MÄRZ 1988

Durch die Verwendungsmöglichkeit eines Personalcomputers zur Programmentwicklung für eine industrielle Applikation entfällt die ansonsten notwendige Anschaffung eines eigenen Entwicklungssystems (VME, ECB usw.).

Durch konsequente 19"-Technik (zum Einsatz kommen Platinen im Einfach- und/oder Doppel-Euro-Format) sind ohne großen Aufwand beliebige Schutzarten nach DIN 40 050 realisierbar.

INDUTRONIC
Dipl.-Ing. Kreiger GmbH & Co. KG
D-8011 Poing/München
Gruberstraße 46
Telefon (0 81 21) 7 01-0
Telefax (0 81 21) 7 13 64

INHALT

INFO

Ray Duncan: Wir in den USA ... 6
Der Top-Computer-Spezialist aus den USA berichtet über Neues und Interessantes aus der amerikanischen Computer-Szene

Transputer-Module 52

Neues zur Büroautomation von Data General 68

Der Know-How-Computer 115

TEST

Zenith Z-386 mit OS/2 40
Mit dem Z-386 kommt ein Rechner mit der CPU 80386 auf den Markt, der ein Erfolg werden muß. Er ist modular aufgebaut, schneller als viele vergleichbare Computer, ungewöhnlich preisgünstig und kommt mit einem Super-Monitor

In Farbe: Schneiders EGA-PC 43
Ungewöhnlich klein ist der EGA-PC von Schneider, den wir in einer Konfiguration mit Festplatte getestet haben

Devpac – Assembler der Superlative 45
Auf dieses Programm haben alle Atari-ST-Anwender gewartet, die gerne schnell und komfortabel Programme in Maschinensprache erstellen wollen. Devpac ist ein Editor mit integriertem Assembler

Turbo-Pascal wird professionell 50
Daß man gute Sachen noch besser machen kann, beweist Borland mit der neuesten Version von Turbo-Pascal. Das Fenster- und Menükonzept ist fast so wie bei Turbo-C. Eine Online-Hilfsfunktion hilft dem Einsteiger

Quick-C von Microsoft 53
Bislang waren C-Compiler äußerst kompliziert zu bedienen. Daß es auch einfacher gehen kann, zeigt Microsoft mit Quick-C. Die Bedienung über Pull-Down-Menüs und die hohe Compiler-Geschwindigkeit machen das Programmieren in C zur Freude

Grafik mit der VEGA VGA 58
Genau 29 Betriebsarten unterstützt die Grafikkarte VEGA VGA. Ihre größte Auflösung erreicht die Karte mit 800 × 600 Punkten bei 16 Farben. Ein Artikel mit viel Hintergrundinformationen und Programmiertips

C-Compiler im Vergleich 62
Für den Atari ST werden mittlerweile zahlreiche C-Compiler angeboten. Ein Insider, der lange Zeit mit verschiedenen Programmen Erfahrungen sammeln konnte, schildert, worauf man bei der Auswahl achten muß. Drei C-Compiler werden ausführlich vorgestellt

Super-Tabellenkalkulation 69
Ein Programm für alle, die umfangreiches Zahlenmaterial mit dem PC untersuchen wollen, ist Quattro von Borland. Es ist einfach zu bedienen und hilft mit seinen leistungsfähigen Rechenfunktionen, schnell zu einem aussagekräftigen Ergebnis zu kommen

GRUNDLAGEN

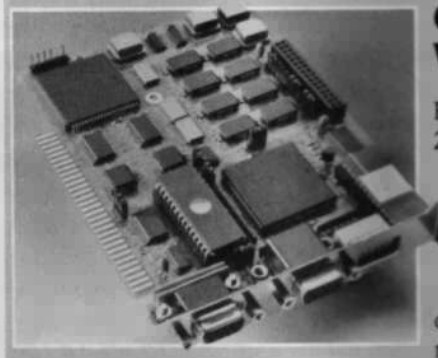
Was man über Modula wissen sollte 71
Die derzeit populäre Programmiersprache Modula-2 wird grundsätzlich und verständlich vorgestellt. Auf die Parallelen zu Pascal wird hingewiesen. Unterschiedliche Typ-Deklarationen werden mit Beispielen erklärt

Titelfoto: Klaus Hager

Zenith Z-386 mit OS/2

Computer mit der CPU 80386 schießen wie Pilze aus dem Boden. Im Z-386 von Zenith wird der Prozessor mit 16 MHz getaktet. Bei unserem Vergleich mit anderen 80386-Rechnern fanden wir einiges Interessantes heraus. Der Z-386 kann mit MS-DOS und mit OS/2 laufen und verfügt über den besten Video-Monitor, den es derzeit im Massenmarkt zu kaufen gibt. Interessant ist der Vergleich von Programmen unter MS-DOS und der Kompatibilität von OS/2.

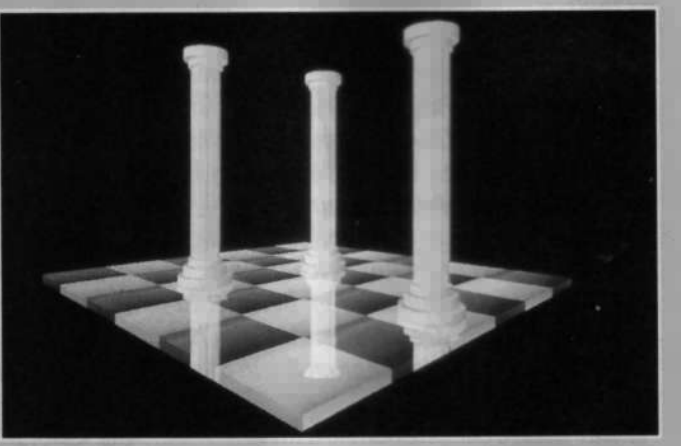
Seite 40



Grafik mit der VEGA VGA

Für die Computer der PS/2-Serie hat IBM außer der Mikro-Kanal-Architektur einen neuen Farbgrafikstandard geschaffen: VGA. Mit einer Karte nach dem VGA-Standard kann man je nach Betriebsart bis zu 256 Farben gleichzeitig darstellen. Über den VGA-Standard hinaus geht die VEGA VGA: 28 weitere Betriebsarten werden von dieser Karte beherrscht. Wir stellen die Steckkarte vor, erklären auch technische Details und geben Programmiertips.

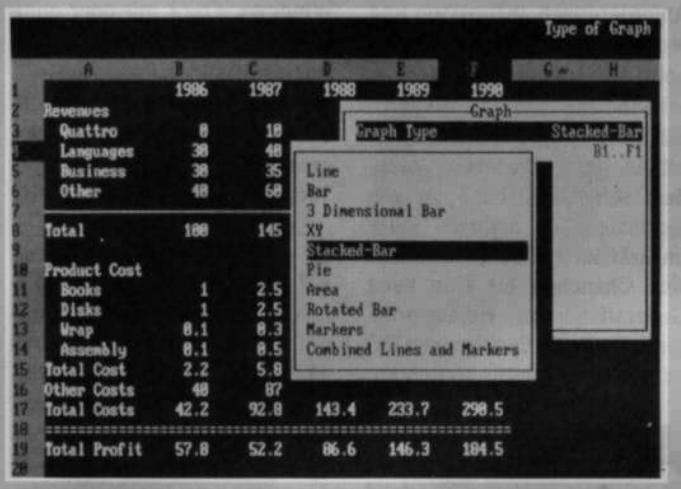
Seite 58



Digitale Bilderzeugung und -verarbeitung

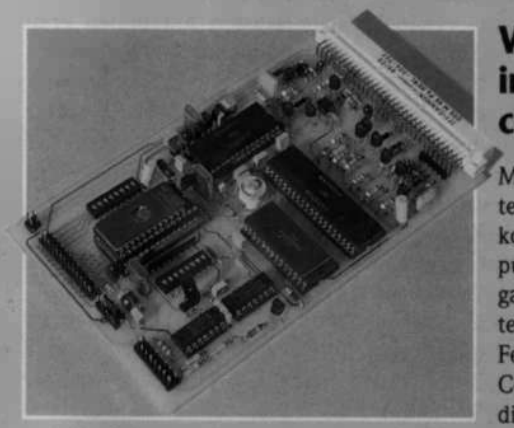
In den letzten Jahren hat die Computergrafik auf PC durch immer leistungsfähigere Grafikprozessoren gewaltige Fortschritte gemacht. Einen der modernsten, den Intel GDP 82786, zeigt mc in einer kompletten Applikation am mc-68000-Computer – denn ein Beispiel sagt mehr als tausend Worte.

Seite 80



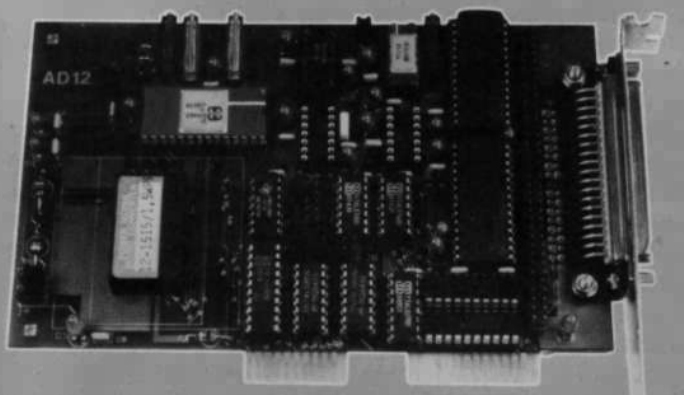
Super-Tabellenkalkulation

Tabellenkalkulation heißt ein Zauberwort der mit PC arbeitenden Insider. Dabei handelt es sich aber „nur" um die Auswertung von Zahlenreihen. Sehr komfortabel und schnell kann man Zahlen jetzt mit Quattro von Borland jonglieren. Die einfache Erzeugung von Grafiken aus dem vorliegenden Zahlenmaterial ist nur eine Stärke dieses interessanten Programmes. **Seite 69**



Videotext in den Mikrocomputer

Mit dem mc-Videotext-Decoder bekommt jeder Computeranwender Zugang zum Videotext. Zwischen dem Fernseher und dem Computer werden die Daten über die serielle Schnittstelle transportiert. In dieser Ausgabe wird das Medium Videotext vorgestellt. Dabei gehen wir besonders auf die technischen Details ein. **Seite 126**



Professioneller A/D-Umsetzer für den PC

Viele PC-Anwender haben früher oder später den Wunsch, Umweltdaten mit dem PC zu erfassen und auszuwerten. In der Regel werden die Daten, wie z. B. Temperatur oder Helligkeit, als analoges Signal von den Sensoren geliefert. Die Umsetzung der analogen Signale in digitale Impulse, die der PC bearbeiten kann, übernimmt ein A/D-Umsetzer. Wir stellen eine Steckkarte vor, die jeder selbst bauen kann. **Seite 104**

INHALT

Dreifach genäht **76**
In diesem Artikel geht es um die Codierung von Informationen. Wir stellen einen Algorithmus vor, der optimale Sicherheit bei der Verschlüsselung von Daten verspricht

Digitale Bilderzeugung und -verarbeitung **80**
Am Beispiel einer Grafikkarte für den mc-68000-Computer wird der praktische Einsatz des Intel GDP 82786 demonstriert

Prolog und Logik **87**
In mc steht, wie der Beweismechanismus von Prolog funktioniert: Formale Logik im Computer praktisch absolviert

SOFTWARE

Wecker im PC **90**
Damit man einen wichtigen Termin nicht verpaßt, kann man sich einen Knoten ins Taschentuch binden oder einen Wecker stellen. Wenn dieser Wecker im PC neben den Anwenderprogrammen läuft, kann man beruhigt bis zur eingestellten Zeit durcharbeiten

Zeichnen von Karikaturen **94**
Karikaturen zu zeichnen kann mit dem Amiga vereinfacht werden. Wir bringen einen Vorschlag für ein Basic-Programm

Die Zeit ist reif **99**
Zur Programmierung zeitgesteuerter Vorgänge dient das Turbo-Pascal-Programm, das wir hier vorstellen. Die Routine bietet fünf unterschiedliche Timer-Typen zur Auswahl an. Impulsdiagramme helfen bei der Auswahl des richtigen Typs

HARDWARE

Professioneller A/D-Umsetzer für den PC **104**
Ein Leckerbissen für alle PC-Anwender, die mit ihrem Computer eine universelle Schnittstelle zur Messung von Umweltdaten haben wollen

Der mc-modular-AT **116**
Im letzten Teil unserer Beschreibung des Selbstbau-ATs geht es um das Betriebssystem MS-DOS. Schwerpunkte liegen bei der richtigen Behandlung von Disketten und Festplatten, bei der Bedienung von EDLIN und dem Konzept von Batch-Dateien

mc-EPROM-Brenner für Z80-Systeme **125**
Vor über einem Jahr stellten wir ein universelles EPROM-Programmiergerät vor. Die starke Nachfrage nach dem mc-EPROM-Brenner führte zu einer Anpassung an Z80-Systeme

Videotext in den Mikrocomputer **126**
Mit dem mc-Videotext-Decoder können die Videotexttafeln über eine serielle Schnittstelle zur weiteren Bearbeitung in jeden Computer geladen werden

RUBRIKEN

Editorial **3**
Briefe **8**
Bücher **38**
Zitat des Monats **44**
Spruch des Monats **60**
Markt **140**
Vorschau **182**
Impressum **8, 181**

RAY DUNCAN

Wir in den USA ...

Hallo, liebe mc-Leser. Sie haben meinen Namen bisher nicht in mc gelesen. Deshalb möchte ich mich und diese neue Kolumne zunächst vorstellen.

Ich bin ein Mann, der Mikrocomputer seit den Tagen der Imsai- und Altair-Modelle programmiert und von Anfang an auch darüber schreibt. Als die Mikrocomputer kamen, da war mir, als wäre ich gestorben – und im Paradies wieder aufgewacht. Denn vorher, da habe ich mit Mini-Computern gearbeitet und mußte die CPU und die Disketten-Laufwerke mit anderen Kollegen teilen. Ich habe in den Zeitschriften „Dr. Dobb's Journal" und „Softalk/PC" regelmäßig geschrieben. Zur Zeit läuft unter dem Titel „Power Programming" eine ständige Rubrik von mir über C und MASM im amerikanischen „PC-Magazin". Mein erstes Buch „Advanced MS-DOS" wurde 1986 von Microsoft Press veröffentlicht. Es ist im Vieweg-Verlag jetzt auch in deutscher Übersetzung erschienen. Zur Zeit arbeite ich an dem Buch „Advanced OS/2", das von Microsoft Press im Frühjahr 1988 veröffentlicht werden wird. Gerade aus der Druckerei gekommen ist das Buch „MS-DOS Encyclopedia", das ich redigiert habe; es ist in den Buchläden ab Januar zu haben. Neben meiner Tätigkeit als Autor von Büchern und Zeitschriftenartikeln arbeite ich in meinem eigenen kleinen Software-Haus (Laboratory Mikrosystems Inc., kurz LMI). Neben einer Spezialisierung auf Forth-Interpreter und -Compiler programmieren wir auch in Assembler, C und jeder anderen Programmiersprache, die zur Lösung einer Aufgabe nützlich ist.

Hier in dieser Kolumne will ich nicht über die technischen Details des Programmierens berichten, denn dieses Gebiet decken die deutschen mc-Autoren bestens ab. Ich möchte vielmehr mit Ihnen die Trends und Tendenzen in der Software-Industrie diskutieren, so wie wir sie in den USA sehen, und dabei vor allem die Gewichtung auf Betriebssysteme und Systeme zur Programmentwicklung legen. Gerade jetzt beginnt ein sehr interessanter Abschnitt in der Evolutionsgeschichte der Mikrocomputer und der zugehörigen Software. Der Markt wurde nämlich in den drei Jahren seit Erscheinen des IBM-PC/AT zunehmend langweilig und stagnierte. Beinahe jedes wichtige Softwaregebiet im Markt wurde von jeweils einem Anbieter dominiert: Die Sprachen und Betriebssysteme kamen von Microsoft, die Datenbanken von Ashton-Tate, die Spreasheets von Lotus und so fort. Der gemeinsame Nenner in Bezug auf die Hardware war der PC mit dem 8086/88 im Gehäuse. 80286-Maschinen im klassischen „real mode" galten als „power machines".
1987 gab es aber nun Anzeichen für einen durchgreifenden Wandel. Die 80386-Rechner-Typen begannen den Markt zu erobern und wurden so schnell preiswert, daß alle Prognosen der Marktbeobachter Lügen gestraft wurden. Zugleich etablierte sich der Macintosh von Apple, als er in Gestalt des SE und des Modelles II mit Erweiterungsplätzen, großem Speicher und Numerik-Coprozessor sowie mit hervorragender Farbgrafik erschien, als zuverlässige und brauchbare Maschine für den Geschäftsmann. Das Augenmerk der Computerbenutzer richtete sich deshalb auf die Querverbindungen zwischen Mac und PC, einmal hinsichtlich der Portabilität der Anwendersoftware, dann aber auch wegen der Datei-Kompatibilität und der Vernetzbarkeit der Systeme.

Der große Krieg aber ist auf dem Software-Feld entbrannt. Die Übermacht von Lotus 1-2-3 auf dem Markt der Spreadsheets scheint sich zu lockern, wie vormals die von Visicalc. Die Ironie dabei: Die Autoren von Visicalc versäumten es damals, das Produkt immer wieder den gestiegenen Anforderungen anzupassen und wurden vom Markt gefegt, als Lotus 1-2-3 mit überlegener Geschwindigkeit, mit Funktions-Bibliothek und mit Makros erschien. Jetzt wiederum, nachdem seinerseits 1-2-3 im wesentlichen für mehrere Jahre unverändert blieb, eröffnen sich Chanchen für eine neue Generation von Herausforderern. Borlands Quattro und PC Excel von Microsoft greifen an.

Auf dem Gebiet der Programmiersprachen ist der heiße Wettbewerb zwischen Borland und Microsoft für uns alle von großem Nutzen. Nach der Eroberung des Pascal-Marktes bedrängt Borland jetzt mit seinem preiswerten und leicht zu handhabenden Turbo-C-Compiler das Flaggschiff von Microsoft. Im Gegenzug hat Microsoft seinen C-Compiler mit Version 5.0 drastisch verbessert, hat mit Quick-C eine Programmierumgebung mit Grafik-Unterstützung, mit Debugger und Help-Menüs für interaktives Arbeiten geschaffen und hat sein Quick-Basic in zwei Jahren dreimal größeren Revisionen unterworfen. Was die Betriebssysteme betrifft, da gibt es im Augenblick nur Verwirrung und Durcheinander. OS/2, Microsofts Multitasking-System mit virtueller Speichertechnik für 80286-Rechner ist nun wirklich da, wird aber viel kritisiert („Zu spät, zu klein..."). Die Software-Entwickler haben hier längst den -386er im Visier, auf dem OS/2 zwar läuft, auf dem es aber nicht den 32-Bit-Modus und nicht die 4 Gigabyte Segmente regieren kann. Zwei Jahre soll es wohl noch dauern, bis eine 80386-Version kommt. Mittlerweile scheint sich der Betriebssystem-Markt böse aufzusplitten. MS-DOS wird wahrscheinlich zusammen mit leidlich verbesserten Windows (grafische Benutzeroberfläche) Standard für 8086/88-Maschinen bleiben. Auf 80286-Maschinen wird OS/2 parallel zu MS-DOS erst Fuß fassen, wenn es Programme gibt, die wirklich die Fähigkeiten von OS/2 nutzen. Für die 80386-Maschinen kann man sich viele Szenarios denken. MS-DOS mit oder ohne Windows, OS/2 mit oder ohne Presentation Manager, Multitasking-80386-Software-Systeme wie DesqView und Windows/386, Unix-Derivate und ganz andere Systeme wie VM/386, PC-MOS/386 und Qunix.

Soweit aus den USA. Obwohl ich Ihnen mit meiner Kolumne möglichst exakte Analysen und Tendenzen mitteilen möchte, hoffe ich doch sehr, daß ich durch den Kontakt mit Ihnen und dem europäischen Markt in die Lage versetzt werde, meinen amerikanischen Lesern einiges an Empfinden für die außeramerikanischen Geschehnisse in der Computerwelt nahezubringen. Während verschiedener Aufenthalte in Deutschland mußte ich erfahren, daß die Europäer von den meisten amerikanischen Software-Häusern mit hohen Preisen und wenig Service bedacht werden. Neue Versionen werden erst sechs Monate später als in den USA angeboten, der Preis fürs Aktualisieren einer Version ist höher als der Preis für ein Original in den USA und von den Umlauten, den Formaten von Datum und Tageszeit in Programmen sei hier erst gar nicht gesprochen. Was mich ebenfalls bekümmert, ist die Ignoranz, die amerikanische Journalisten und Software-Entwickler oft an den Tag legen, wenn es um ausländische Produkte geht. So glaubt man in der amerikanischen Szene fest an die Story über Phillip Khane, der in einem Jahr, Nacht für Nacht ohne Schlaf, in seiner Garage Turbo-Pascal geschrieben haben soll, wobei er all sein Kapital riskierte. In Europa dagegen scheint es bekannt zu sein, daß Turbo-Pascal nur eine andere Verpackung eines dänischen Pascal-Compilers ist. Vielleicht kann diese Kolumne die gegenseitige Kommunikation beider Welten verbessern. Schreiben Sie mir, was Sie an den USA interessiert; entweder an mc oder direkt an mich.

Meine Adresse:
Ray Duncan, c/o Laboratory Microsystems Inc., 3007 Washington Blvd., Suite 230, Marina del Rey, CA 90292.

Summa summarum: für alle eine lohnende Investition!

Egal, welche Anforderungen Sie an einen Drucker stellen, der neue FUJITSU DL 3300/3400 nimmt Ihnen den Druck! Denn diese universellen Leistungsdrukker lösen jede Aufgabe: schnell, perfekt und in bestechender Qualität. Sie sind auf höchste Zuverlässigkeit konzipiert und stehen mit 8000 Betriebsstunden MTBF in ihrer Klasse an der Spitze.

Die neuen 24-Nadel-Matrixdrucker sind für einfachen Einsatz von IC-Karten für z. B. größere Schriftvielfalt und erweiterte Zeichensätze ausgelegt und für Farboption nachträglich aufrüstbar.

Sie bieten – wie alle FUJITSU-Drucker – ein Höchstmaß an Bedienungskomfort und ermöglichen einen problemlosen Wechsel von Endlos- auf Einzelblattbetrieb.

Beste FUJITSU-Technologie also. Und das zu einem Preis, der bei diesem Leistungsspektrum seinesgleichen suchen dürfte.

Bitte senden Sie mir genauere Informationen über

☐ den FUJITSU DL 3300/3400

☐ das gesamte FUJITSU-Drucker-Programm

mc 3/88 SC

Name _____

Straße _____

PLZ/Ort _____

Firma _____

Coupon an: FUJITSU Deutschland GmbH
Rosenheimer Str. 145 · 8000 München 80
Drucker-Telefon-Hotline 089/41 30 11 52

FUJITSU

Drucker von Japans Computerhersteller Nr. 1

Die Mikrocomputer-Zeitschrift

Franzis-Verlag GmbH
Karlstraße 37–41, 8000 München 2
Postfach 37 01 20, 8000 München 37
Telefon (0 89) 51 17-1
Telex 5 22 301
Telefax (0 89) 51 17-3 79
Tedas (0 89) 59 84 23 und 59 64 22

Chefredakteur:
Dipl.-Math. Ulrich Rohde

Stellvertretender Chefredakteur:
Dipl.-Ing. (FH) Ulrich Kruppe

Redaktion:
Redaktionsassistentin:
Rita Schleser (0 89) 51 17-3 54
Ressortredakteure: Horst Brand, Dipl.-Ing. (FH) Wolfgang Hascher (fl), Aktuelles: Reiner Schönrock, PC-Technik: Dipl.-Ing. (FH) Dieter Strauß

Korrespondent: USA:
Ray Duncan

Ständiger Mitarbeiter:
Dipl.-Ing. (FH) Herwig Feichtinger (zu erreichen unter der Anschrift der Redaktion)

Art Direktion:
Dipl.-Designer (FH)
Volker Hilbel

Layout, Grafik, Herstellung:
Josef Wurzinger

Franzis-Labor:
Neucom electronic GmbH

Software Service:
SHAMROCK Software Vertrieb
(0 89) 59 54 68

Verantwortlich für den Inhalt:
Dipl.-Math. Ulrich Rohde

Verantwortlich für Anzeigen:
Dietger Kötter

Gesamtherstellung:
Franzis-Druck GmbH
Karlstraße 35
8000 München 2
Telefon (0 89) 51 17-1

© 1988 für alle Beiträge bei Franzis-Verlag GmbH

ISSN: 0720-4442

Vertriebskennzeichen:
B 7745 E

Sonderdrucke:
Jakob Wintersberger

Lizenzen:
Georg Geschke, Königstr. 8,
3000 Hannover 1,
Telefon (05 11) 34 53 62

Urheberrechte:
Die in mc veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, insbesondere das der Übersetzung in fremde Sprachen, vorbehalten. Auch die Rechte der Wiedergabe durch Vortrag, Funk- oder Fernsehsendung im Magnettonverfahren oder ähnlichem Wege bleiben vorbehalten. Fotokopien für den persönlichen oder sonstigen eigenen Gebrauch dürfen nur von einzelnen Beiträgen oder Teilen daraus als Einzelkopien hergestellt werden.

BRIEFE

Potenzen in Pascal

Im Artikel (mc 12/87, S. 84) wird angegeben, daß $0^0=1$ sei. Nun ist aber 0^0 ein „unbestimmter Ausdruck" wie z. B. auch 0/0. Dies bedeutet, daß der wahre Wert für diesen Ausdruck je nach Fall bestimmt werden muß. Ein konstruiertes Beispiel:

$$\lim_{x\to 0} x^{1/\ln(x)} = ?$$

Da ln(x) für $x\to 0$ gegen ∞ strebt, ist obiger Ausdruck ein unbestimmter Ausdruck der Form 0^0.
Da gilt $x = e^{\ln(x)}$, kann umgeformt werden:

$$\lim_{x\to 0} x^{1/\ln(x)} =$$

$$= \lim_{x\to 0} e^{\ln(x) * 1/\ln(x)} =$$

$$= e^1 = e \neq 1$$

Wer will, kann dies auch mit dem Taschenrechner ausprobieren. Solange für x eine Zahl > 0 eingesetzt wird, ist das Resultat e. Wer's nobel will, kann auch den Computer benützen:

```
10 Input x:if x<=0 then 10
20 Print x (1/ln(x))
30 Goto 10
```

Patrick Feisthammel
Zürich

Flott und mit Struktur

In Ihrem Artikel (mc 10/1987) schreiben Sie, daß Borlands Turbo-Basic kompatibel zum GW-Basic von Microsoft ist. Auch Borland behauptet dies. Ich mußte leider die Erfahrung machen, daß dem nicht ganz so ist. Bereits das erste GW-BASIC-Programm, das ich mit Turbo-Basic übersetzen wollte, verweigerte den Dienst: Der GW-Basic Befehl VIEW-PRINT ließ sich nicht verarbeiten.
VIEW PRINT ist sehr nützlich, um Bereiche des Bildschirmes, z. B. die Kopf- und Fußzeile, gegen Löschen (mit CLS) oder Überschreiben zu schützen. In vielen meiner Programme nutze ich diesen Befehl für Überschrift und Copyright. Microsofts Quickbasic verarbeitet VIEW PRINT einwandfrei.

G. Ohland
4790 Paderborn-Sande

Kooperation ist alles

Im Beitrag (mc 11/87, Seite 84) wurden die Co-Routinen als PROCESS-Typ deklariert. Dies wurde durch Nikolaus Wirth in der neueren Syntax von Modula-2 geändert. Nun müssen Co-Routinen als ADRESS-Typ deklariert werden. Die entsprechenden Zeilen lauten:

```
MODULE wuerfelSpiel;

FROM SYSTEM IMPORT
  WORD, ADR, SIZE,
  ADDRESS, NEWPROCESS,
  TRANSFER;

(*...*)

VAR Spieler1, Spieler2,
  Haupt : ADDRESS;
  (* Co-Routinen-Variable *)

(*... Rest wie gehabt *)
```

Christoph-M. Lehmann
7909 Tomerdingen

In Sachen Listing

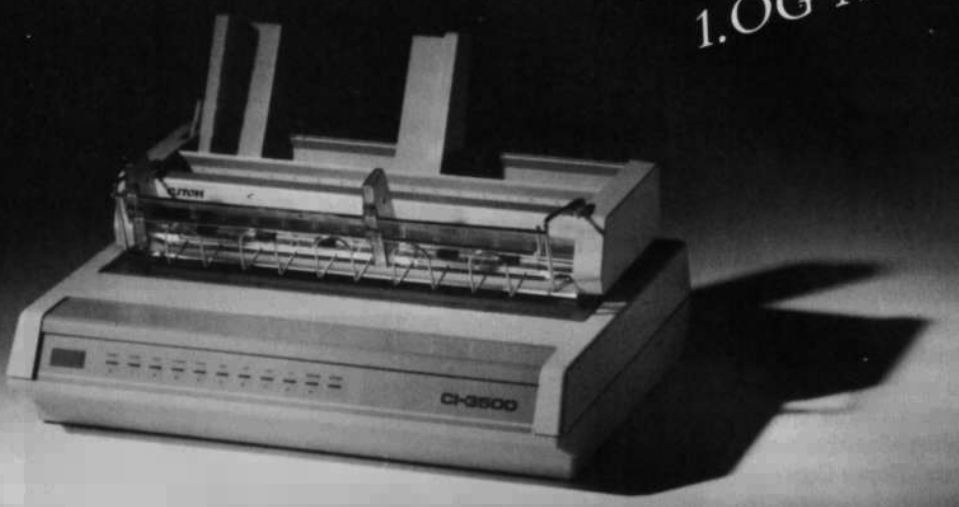
Sehr gerne nehme ich Ihre Anregung an, meine Meinung zu dem Artikel „ANMERKUNGEN" in mc 12/87, S. 78 abzugeben. Auch mich bringen Programmauflistungen, vor allem solche in „C", oft in Verzweiflung. Die schon fast an Lynchjustiz grenzende Kritik an dem Programm aus mc 4/87 halte ich nur in einem einzigen Punkt für gerechtfertigt. Die verkleinerten, mit Matrixdruckern erstellten Programmtexte erhöhen die Schwelle zum Abschreiben gewaltig. Eine Art impliziter Kopierschutz vielleicht? Die Abhandlung über die Zuweisung einer binären NULL schießt etwas über das Ziel hinaus. Syntaktisch halte ich es für völlig gleichwertig, '0', 0x00 oder 0 zu schreiben, der Unterschied liegt einzig in der Basis (oktal, hexadezimal oder dezimal). Kniffliger wird das Problem bei der philosophischen Frage nach „GOTO". Die vom Autor angebotene Variante ist wahrscheinlich noch unlesbarer als das unelegante Original. In der Testphase eines neuen Programms kommt bei mir hin und wieder auch so manches „GOTO" vor, im Sinne einer besseren Pflegbarkeit verschwinden die meisten davon aber wieder. Die resistentesten „GOTOs" waren aber immer die in „FOR"-Schleifen, das gebe ich auch zu. Für Programme, die auch einen gewissen ditakischen Wert haben sollen, wähle ich seit einiger Zeit den im *Bild* gezeigten Weg. Der Trick bei der Sache ist der, daß die Laufvariable nur dann den Wert „max" annehmen kann, wenn die Schleife nicht mit „BREAK" verlassen worden ist. Ich finde es sehr gut, daß Sie in Ihrer Zeitschrift schön langsam von den seitenlangen Auflistungen von BASIC-Programmen abgehen und immer mehr Programme in Sprachen bringen, die auch Rückschlüsse auf Algorithmen und Datenstrukturen zulassen.

Dr. Harald Pollack
Wien

```
for (i = 0;i < max;i++)
     {
     .
     if (Übereinstimmung) break;
     .
     }
if (i == max)
     {
     printf("keine Übereinstimmung");
     return;
     }
printf("Übereinstimmung nach %d Durchgängen",i+1);
return;
```

Damit sollten Sie Ihren Computer mal erleben

CeBIT
HALLE 4
1.OG A61



CI 3500 von C.ITOH

Als kleinstes Mitglied der Schnelldrucker-Familie schafft dieser robuste Nadeldrucker 350 Zeichen pro Sekunde bei 10 cpi, ganze 7,5 Sekunden sind es, die er für den Dr.-Grauert-Text braucht. Seine Emulationen DEC LA 100, IBM Proprinter, IBM 3287 Koax. Die Produktfamilie wird ergänzt durch die Zeilendrucker CI 300+ und CI 600+. Sie arbeiten nach einem neuen Shuttle Bank Prinzip, das höchste Schriftqualität in jedem Tempo garantiert. Im Korrespondenzdruck leistet der CI 300+ 80 Zeilen pro Minute, im High Speed Mode bringt es der CI 600+ sogar auf 800 Zeilen pro Minute. Was Grafiken angeht, zeigen sie auch hier schön schnell, daß sie vieles können: OCR-Schriften, Barcodes, Großschrift, Geschäftsgrafiken.

PRINT PARTNER

Die Drucker-Spezialisten

Berlin
INFORDATA Datentechnik GmbH Berlin
Eisenacher Straße 123, 1000 Berlin 30
Tel. (030) 216 50 50

Hamburg, Bremen, Schleswig-Holstein
TEXTCOM Mikrocomputer-Vertrieb GmbH
Steindamm 32, 2000 Hamburg 1
Tel. (040) 24 36 16, Telex 214768

Niedersachsen
TEXTCOM Mikrocomputer-Vertrieb GmbH
Geschäftsstelle Hannover
Landwehrstraße 61, 3000 Hannover 81
Tel. (05 11) 83 09 00

Nordrhein-Westfalen
AC-COPY Datentechnikvertrieb GmbH
Kurbrunnenstraße 30, 5100 Aachen
Tel. (02 41) 50 60 96, Telex 832368

Hessen, Rheinland-Pfalz, Saarland
PRINTEC Datentechnik Vertriebs-GmbH
Dieselstraße 20,
6452 Hainburg 2
Tel. (0 61 82) 7 78-0, Telex 4184055
Teletex (17) 6182915

Nordwürttemberg/Nordbaden
TEACH Hard- und Software-Vertrieb GmbH + Co.
Siemensstraße 22, 7257 Ditzingen 1
Tel. (0 71 56) 30 01-0, Telex 7245248

Südwürttemberg/Südbaden
Binder Datentechnik GmbH
Geschäftsstelle VS-Villingen
Mönchweilerstraße 1, 7730 VS-Villingen
Tel. (0 77 21) 88-0, Telex 792568,
Fax (0 77 21) 88-359

Bayern
Binder Datentechnik GmbH
Geschäftsstelle München
Konrad-Celtis-Straße 81, 8000 München 70
Tel. (089) 7 14 60 81, Telex 522438

Dieter Strauß

Zenith Z-386 mit OS/2

Modular aufgebaut und mit OS/2, das ist der Z-386. Er kann aber auch das normale MS-DOS verwenden. Dieser Rechner ist das Spitzenprodukt von Zenith für alle, die große Rechnerleistung benötigen. Der zugehörige Monitor ZCM-1490 überrascht mit einer guten Bildqualität.

Das System Z-386 (*Bild 1*) ist als Workstation für anspruchsvolle Benutzer in der Industrie, Wissenschaft und Technik gedacht. Es ist mit einem 80386-Prozessor bestückt, der mit 16 MHz getaktet wird.
Die Busplatine hat Platz für zehn Steckkarten, von denen bei folgender Bestückung noch vier Steckplätze frei bleiben:

- CPU-Karte
- zwei 1-MByte-Speicherkarten
- Festplatten-/Floppy-Controller
- E/A-Karte mit paralleler und serieller Schnittstelle
- Grafikkarte

Bestehen hohe Anforderungen an den Systemdurchsatz kann man die Workstation mit einer Cache-Karte bestücken. Sechs Steckplätze sind direkt mit dem 32-Bit-Systembus verbunden. Neben den Steckverbindern befinden sich auf der Busplatine nur noch die Batterie, der Uhrenbaustein 146818, einige Treiber- und PAL-Bausteine und ein 14,31818-MHz-Quarzoszillator.

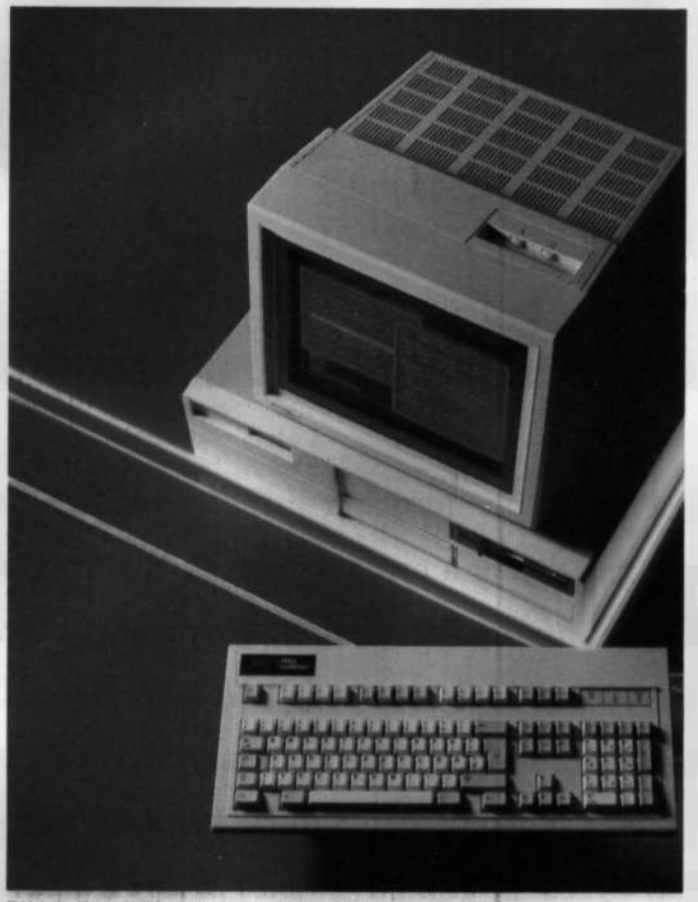


Bild 1. Der Z-386 ist ein modularer 386-Rechner

Im Gegensatz zu den Rechnern der AT-Klasse hat sich bei den 386-Rechnern noch kein einheitliches Pinout für die 32-Bit-Steckplätze durchgesetzt. Es ist auch fraglich, ob dies geschehen wird, da IBM zur Mikrokanal-Architektur übergegangen ist. Zum Glück sind die PC- und die AT-Steckplätze bei 386-Rechnern kompatibel geblieben.
Die 32-Bit-Steckplätze sind so konstruiert, daß man auch PC- und AT-Karten einstekken kann, wenn diese hinter dem Steckkontakt nicht heruntergezogen sind.
Interessant ist die CPU-Karte, da sie wahlweise mit einem 80287- oder einem 80387-Numerik-Prozessor bestückt werden kann. Als Spezialität kann man den System-Monitor-ROM auf der CPU-Karte bezeichnen, auf dessen Bedeutung wir später eingehen. Auf der CPU-Karte befindet sich weder der Arbeitspeicher noch das BIOS. Beide sitzen auf einer Speicherkarte, die mit 1 MByte RAM bestückt ist. Auf die Speicherkarten greift der Prozessor mit 32 Bit zu.
Wie beim mc-modular-AT ist der Festplatten- und der Floppy-Controller auf einer Platine untergebracht. Auf der E/A-Karte befindet sich der Parallel-Port und der serielle Port. Hilfreich für den Service-Techniker ist die Leuchtdioden-Reihe, die CPU-, ROM-, RAM,- Interrupt-Controller- und Massenspeicher-Fehler anzeigt.
Nur die Grafikkarte ist in einem PC-Steckplatz untergebracht. Sie ist kompatibel zum VGA-Standard und ist mit dem EGA-Chipsatz von Chips & Technologies bestückt, einem kundenspezifischen Schaltkreis, der für die VGA-Eigenschaften zuständig sein dürfte, und der Farbpalette IMSG170S von INMOS. Das BIOS stammt von Sigma Designs.

Insgesamt macht der Aufbau des Computers einen soliden Eindruck. Die Abmessungen des Systems sind fast gleich mit denen anderer 80386-Computer: 53 x 15,5 x 42 cm^3.
Da der Z-386 mit sehr vielen PALs und schnellen Bausteinen der Fast-Familie bestückt ist, entwickelt er relativ viel Wärme. Außer im Netzteil befindet sich neben dem Kartenblock noch ein weiterer Lüfter, dessen Kühlluft mit einem Blech zu den Karten geleitet wird. Nicht jeder 386-Rechner ist so ausgestattet, da jedoch der Z-386 kein Low-Cost-Gerät ist, hat Zenith darauf geachtet, die Gehäuseinnentemperatur niedrig zu halten, um damit die Lebensdauer der Bauteile zu verlängern. „Kühle CMOS-Technik" scheint derzeit bei 386-Rechnern, die mit höheren Taktfrequenzen als AT-Rechner arbeiten, noch nicht möglich zu sein.
Das Interessanteste am Z-386 ist jedoch, daß Zenith ihn wahlweise mit MS-DOS 3.21D oder MS-OS/2 anbietet. Bis zum Zeitpunkt der Veröffentlichung soll sogar DOS 3.3 verfügbar sein.

Integrierter System-Monitor

Wenn sich während des Kalt- oder Warmstarts im Laufwerk A eine Diskette befindet, versuchen Standard-MS-DOS-Rechner von der Diskette zu booten. Da es sich bei Systemen mit Festplatte fast nie um die Systemdiskette handelt, wird man gleich mit einer Fehlermeldung konfrontiert und gebeten, doch bitte die MS-DOS-Systemdiskette einzulegen. Der Z-386 ist da viel freundlicher zum Anwender. Er bootet im-

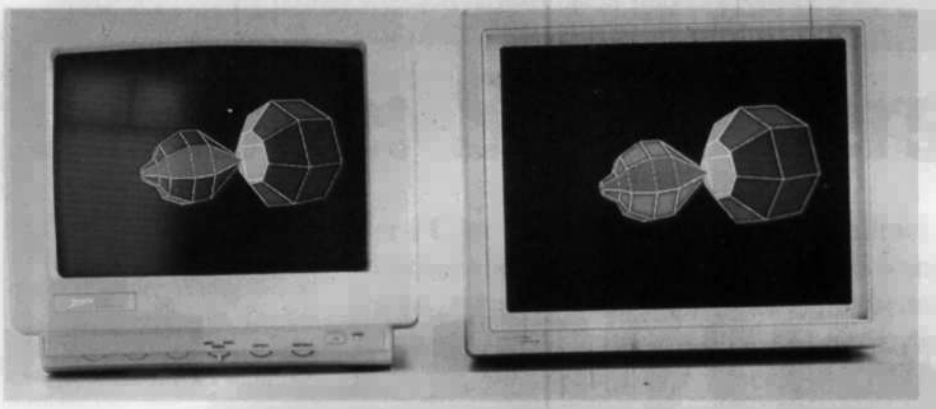


Bild 2. Der linke Bildschirm ist nicht entspiegelt. Reflexionen erschweren die Arbeit. Der rechte Bildschirm ist mit der OCLI-Antireflexionsfolie beschichtet

mer von der Festplatte, auch wenn sich eine Diskette im Laufwerk A befindet. Falls man wirklich von der Diskette booten will, muß man unmittelbar nach dem Kalt- oder Warmstart mit der Tastenkombination Ctrl-Alt-Ins den System-Monitor aufrufen und nach dem Monitorprompt „bf" (Boot Floppy) eingeben. Mit dieser Methode kommt man selbst bei zerstörtem Befehlsinterpreter – der unter MS-DOS COMMAND.COM und unter OS/2 CMD.EXE heißt – in das System.
In dem System-Monitor ist außerdem das Setup-Programm, das Hardware-Testprogramm und der Debugger eingebaut. Obwohl der System-Monitor in einem ROM untergebracht ist, verfügt er über eine rudimentäre Benutzerführung. Gibt man ein Fragezeichen ein, erscheint auf dem Bildschirm eine Liste der zulässigen Befehle. Der Befehlsumfang des Debuggers ist in etwa mit dem des DEBUG.COM vergleichbar.

Reflexionsfreier Bildschirm

Da zu einer Workstation ein ordentlicher Monitor gehört, gibt es zum Z-386 den Farbmonitor ZCM-1490, bei dem ein neues Konstruktionsprinzip – die flach gespannte Lochmaske (Flat Tension Mask) – verwirklicht wurde. Bei diesem Verfahren wird die Lochmaske unter einer mechanischen Spannung von ca. 10 Tonnen pro Quadratzentimeter in den Röhrentubus eingegossen um die Ausdehnung der Lochmaske bei Erwärmung fast völlig auszuschließen.
Das ist wichtig, weil sich in jeder Bildröhre die Lochmaske durch den auftreffenden Elektronenstrahl erwärmt. Wenn sie sich dann ausdehenen würde, würden die Löcher der Lochmaske nach kurzer Betriebsdauer nicht mehr genau über den Farbpunkten liegen. Unschärfen und Farbverschiebungen wären die Folge. Wegen der vorgespannten Lochmaske kann der Monitor ZCM-1490 jedoch mit einer bis zu fünfmal höheren Elektronenstrahlenergie als herkömmliche Monitore arbeiten. Das sieht man der Farbbrillianz und dem Kontrast auch an.
Außerdem ist die Frontscheibe des Monitors mit einer speziellen Folie (OCLI-Folie aus dem amerikanischen Raumfährenprogramm) entspiegelt. Wir konnten uns im Test überzeugen, daß selbst bei direktem Lichteinfall keine Spiegelungen auf dem Bildschirm zu sehen sind (*Bild 2*). Ohne Übertreibung: die Darstellungsqualität des Monitors ist hervorragend.
Bei allen positiven Eigenschaften hat der Monitor auch einen Nachteil: Zur Kühlung seiner Elektronik läuft ständig ein Lüfter mit.

MS-OS/2 einfach installiert

Auf dem Testgerät war bereits MS-DOS 3.21 installiert. Da wir den Rechner auch unter MS-OS/2 testen wollten, mußte dieses erst noch im System eingerichtet werden. Wer jetzt glaubt, daß dazu auf der Festplatte eine neue Partition eröffnet werden muß, der irrt sich gewaltig. OS/2 kann man auf der gleichen Partition wie MS-DOS installieren. Da MS-DOS und OS/2 das gleiche Dateiverwaltungssystem verwenden, kann man ein sogenanntes Dual-Boot-Laufwerk einrichten und dann wahlweise mit einem der beiden Betriebssysteme starten.
Das Einrichten von OS/2 ist sehr einfach: Der Anwender muß lediglich die OS/2-Diskette 1 in das Laufwerk A einlegen und das System starten. Da der Z-386 immer von der Festplatte bootet, muß man das System jetzt mit dem System-Monitor anweisen, von der OS/2-Diskette zu booten. Das Installationsprogramm ist vollständig menügeführt, so daß man kaum etwas falsch machen kann. Mit ihm legt man fest, ob das System zukünftig mit MS-DOS oder OS/2 starten soll. Selbst wenn das System so voreingestellt ist, daß es z. B. mit OS/2 beginnt, kann man es nach dem Warm- oder Kaltstart mit der Alt-Taste veranlassen, mit MS-DOS zu booten. Im Betrieb kann die Voreinstellung jederzeit mit dem SET-BOOT-Befehl geändert werden. Wer kein Dual-Boot-Laufwerk haben will, kann selbstverständlich nur MS-OS/2 auf der Festplatte installieren oder eine zweite Partition anlegen.

Vergleich MS-DOS mit MS-OS/2

Unser Ehrgeiz in der Redaktion war es, den Durchsatz des Rechners unter OS/2 mit dem unter MS-DOS zu vergleichen.
Die größte Hürde, die dazu genommen werden mußte, war die Portierung der beiden mc-Testprogramme PRIM1 und PRIM2 [1] nach OS/2. Zum Glück war die Hürde kleiner als gedacht und die beiden Testprogramme funktionierten zufriedenstellend. Aus Platzgründen können die beiden Programme, die als Kernel-Applikation [2] ausgeführt sind, in dieser Ausgabe nicht abgedruckt werden.
PRIM1 und PRIM2 suchen nach der reinen Rechenvorschrift die ersten 1000 Primzahlen. Sie benutzen nicht den Algorithmus „Sieb des Eratosthenes". Die beiden Programme sind daher, obwohl sie in Assembler geschrieben wurden, relativ langsam. Ziel ist es, das jeweilige Testsystem „ordentlich zu beschäftigen" (und nicht möglichst schnell Primzahlen zu finden). PRIM1 gibt die jeweiligen Primzahlen mit einem DOS-Funktionsaufruf (Funktion 02H: console output) auf dem Bildschirm aus, bei PRIM2 wird dieser Funktionsaufruf einfach übersprungen.
In der MS-OS/2-Version wird die DOS-Funktion 02H durch den OS/2-Betriebssystemaufruf VioWrtTTY (Virtuell I/O Write Teletype) ersetzt. Im ersten Test (*Bild 3*) laufen PRIM1 und PRIM2 unter MS-DOS 3.21D und der Kompatibilitätsbox von MS-OS/2 und werden mit der jeweiligen multitasking-fähigen Kernel-Version verglichen. Die Kernel-Applikation läuft in der aktiven Bildschirmgruppe und es sind keine weiteren Anwenderprogramme aktiv. In der Kompatibilitätsbox ist PRIM1 ungefähr 12 % und als Kernel-Programm ungefähr 23 % langsamer als unter MS-DOS. PRIM2 ist in der Kompatibilitätsbox ungefähr 25 % und als Kernel-Programm ungefähr 18 % langsamer.
Die Testergebnisse muß man vorsichtig interpretieren und keinesfalls verallgemeinern nach dem Motto: „Unter OS/2 sind Ihre Programme 18 % langsamer als unter DOS."

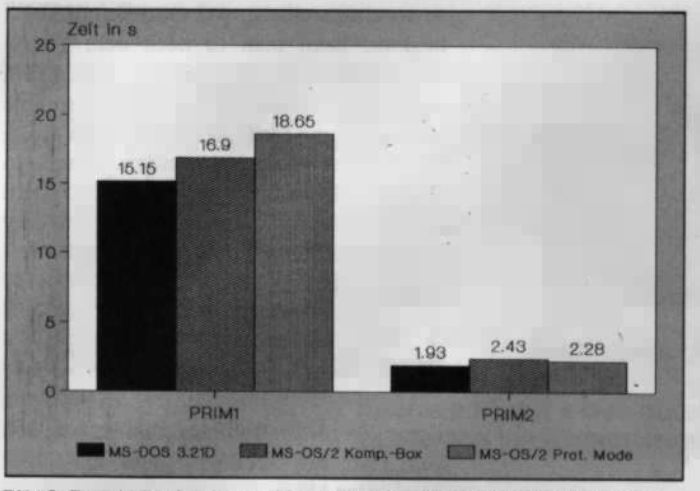


Bild 3. Durch den internen „Verwaltungsaufwand" bei Multitasking laufen vergleichbare Anwendungen langsamer als unter DOS

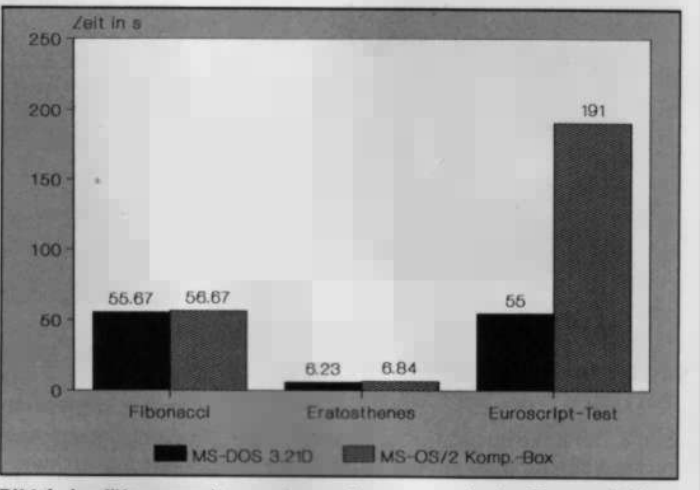


Bild 4. Ausführungszeiten mehrerer Programme in der Kompatibilitätsbox und unter MS-DOS 3.21

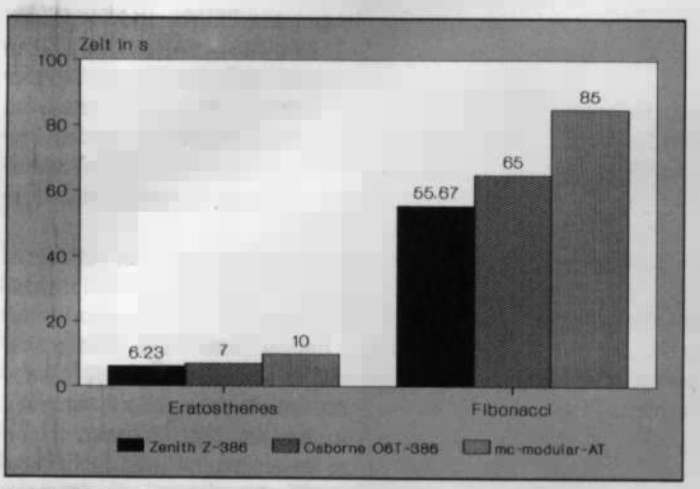


Bild 5. Bei den Benchmark-Programmen „Sieb des Eratosthenes" und „Fibonacci" zeigt der Z-386 was er kann

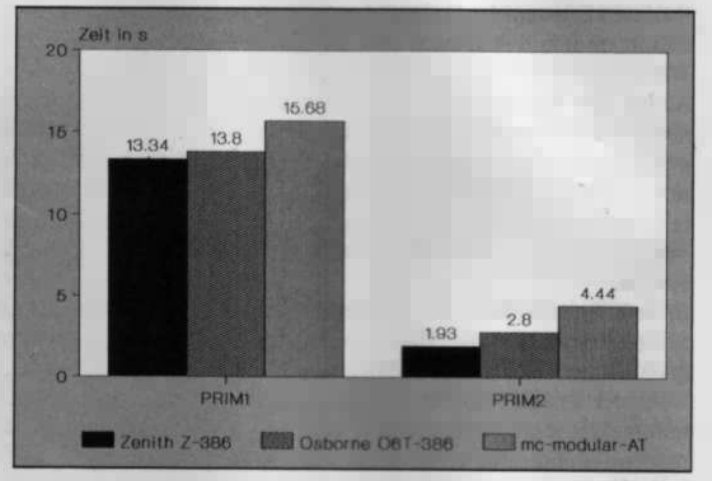


Bild 6. Beim PRIM1-Test wurden alle Testsysteme mit einer Hercules-kompatiblen Grafikkarte bestückt

Besonders die Ergebnisse von PRIM2 zeigen jedoch zwei triviale Sachverhalte:

– MS-DOS-Programme, die in der Kompatibilitätsbox laufen, verlieren Zeit beim Umsetzen der alten Funktionsaufrufe in die neuen. Aus *Bild 4* geht hervor, daß der erhöhte Zeitbedarf stark von der jeweiligen Anwendung abhängt.

– Ein Multitasking-Betriebssystem beansprucht mehr CPU-Zeit als ein einfaches Betriebssystem, da die internen Verwaltungsaufgaben höher sind.

Beim Test mit dem Textverarbeitungsprogramm Euroscript wird im Beitrag „mc-modular-AT" aus mc 2/87 mit dem globalen Ersetzungsbefehl GK der Buchstabe „e" durch „i" ersetzt. Dieser Befehl stellt hohe Anforderungen an die Video-E/A eines Systems, da sämtliche Ersetzungen sofort auf dem Bildschirm gezeigt werden. Gegenüber dem PRIM1/PRIM2-, dem Fibonacci- und dem Test „Sieb des Eratosthenes" [3] ist hier eine deutliche Abnahme des Durchsatzes zu erkennen.

Die verschiedenen Tests zeigen deutlich, daß allgemeine Aussagen zur Verminderung des Systemdurchsatzes im Vergleich MS-DOS mit MS-OS/2 kaum möglich sind. Ein Anwendungs-Programmierer braucht lediglich den Engpaß seiner Applikation zu analysieren, das Programm an der jeweiligen Stellen zu ändern und schon kann das Ergebnis ganz anders aussehen.

Aussagekräftiger für die Leistungsfähigkeit des Z-386 sind *Bild 5* und *Bild 6*. Hier vergleichen wir die Ausführungszeiten verschiedener Programme unter MS-DOS bei mehreren Rechnern. Beim Test PRIM1 geht stark die jeweilige Video-Karte in das Testergebnis ein. Aus dem Zenith Z-386 haben wir daher die Original-Grafikkarte entnommen und eine Hercules-kompatible Grafikkarte eingesteckt.

Obwohl der Zenith Z-386 als auch der Osborne OT-386 mit dem 80386 bestückt sind und beide Computer mit der gleichen Taktfrequenz laufen, ist der Z-386 dennoch schneller.

Die Ursache ist bei der Systemarchitektur zu suchen: der Z-386 greift schneller auf den Speicher zu. Das Testprogramm AT-PERF [3] der amerikanischen Fachzeitschrift PC-Tech-Journal zeigt, daß der Z-386 mit weniger Wartezyklen als der in der letzten Ausgabe getestete Osborne O6T-386 läuft.

Wer einen soliden 80386-Rechner braucht, der durch den modularen Aufbau zukunftssicher ist, und hohe Anforderungen an die Qualität des Monitors stellt, ist mit der Kombination Z-386 und ZCM-1490 für etwas über 13 000 DM sicher gut beraten. Selbstverständlich erfüllt das System alle gängigen TÜV-/GS- und postalischen Vorschriften.

Literatur

[1] *Strauß, D.:* Primzahlen-Sucher. mc 1987, Ausgabe 10, Seite 74.

[2] *Strauß, D.:* Ein erster Blick auf MS-OS/2. mc 1988. Ausgabe 2, Seite 38.

[3] High-Tech Horsepower. Byte. Juli 1987, Seite 101.

[4] *Forgeron, Pierce, Armbrust:* Updating the Evaluation Suite. PC-Tech-Journal. März 1987, Seite 71.

Ulrich Rohde

In Farbe: Schneiders EGA-PC

Der PC ist tot, es lebe der PC-Kompatible! Marktforscher sagen, daß PS/2- und MS-DOS-kompatible Rechner lange, wenn nicht immer, nebeneinander leben werden. Schneiders EGA-PC, ausgerüstet mit Festplatte, ist einer der modernen PCs, die für einen einzelnen „Dienstherren" konzipiert sind.

Der Schneider mit Farben

Die Gretchenfrage bei allen Kompatiblen ist: Spieglein, Spieglein an der Wand, wer ist der kompatibelste im ganzen Land? Da kann man beim Schneider sagen, daß sein physikalischer Aufbau einige Abweichungen besitzt, die anzumerken sind. Positiv ist zu vermerken, daß er wie ein AT von vorneherein eine batteriegepufferte Uhr eingebaut hat, die mit Zusatzspeicher ausgerüstet ist. Der Zusatzspeicher nimmt die Daten von Systemvoreinstellungen auf. Im Handbuch des Schneiders steht, daß er vier freie XT-Steckplätze besitzen würde. Nach dem Öffnen stellte sich heraus, daß es nur drei sind, da ist wohl die Vereinheitlichung der Handbücher ungewollt zu weit gegangen. Die drei Steckplätze können allerdings nur Karten aufnehmen, deren Unterkante nicht herabgezogen ist. Denn der Teil der Hauptplatine, der unter der Abdeckplatte für Steckplätze zugänglich ist, wird durch ein Hochfrequenz-Abschirmblech überdeckt, und zwar gerade so hoch, daß nicht alle auf dem Markt erhältlichen Erweiterungsplatinen problemlos in die Steckplätze passen.

Die Konstruktion besitzt einen entscheidenden Vorteil: die Störstrahlungen des Schneiders sind sehr gering. Das hängt noch mit anderen Details der Konstruktion zusammen. Zum Beispiel ist die Tastatur auf einem dämpfenden Metall-Chassis aufgebaut. Die Signaladern des Verbindungskabels sind über einen Ferrit-Ring als Spule gewickelt, damit Hochfrequenzen aus dem Signal herausgefiltert werden. Das Ergebnis: Ich konnte mit meinem Walkman während der Arbeit am Computer Musik hören. Auch schwächere Sender (UKW, Stereo) wurden nur diskret verrauscht, wenn der Computer lief – und auch nur, sobald ich das Antennenkabel direkt über die Tastatur hängen ließ. Im Wohnzimmer in der Nähe des Fernsehers betrieben, zeigte sich auf dessen Bildschirm eine Einstrahlung, die sich in leichten wellenförmigen Überlagerungen der Bildschirmzeichnung ausprägte. In einem anderen Raum betrieben, gab es keine Schwierigkeiten.

Technik

Der gute alte Norton fördert zutage, daß 640 KByte im Gerät eingebaut sind, daß die CPU und die angeschlossene Peripherie mit dem Faktor 1.9 laufen, was knapp doppelte Rechengeschwindigkeit gegenüber dem Original-XT ergibt. Damit entspricht der Schneider dem guten Standard, den moderne XT-Kompatible einhalten müssen. In ihm verbirgt sich die CPU 8086, er ist ein echter 16-Bit-Rechner. Wie er es mit der Geschwindigkeit hält, das zeigt unser mc-Benchmark „Primzahlen suchen". Das ist ein Programm, das mc-Redakteur Dieter Strauß in Assembler entworfen hat, um den Computern im Test auf den Takt zu fühlen. Es sucht die ersten 1000 Primzahlen, einmal mit Bildschirmausgabe während des Suchens, einmal ohne. Der Zeitunterschied gibt den Anteil an der Gesamtgeschwindigkeit wieder, den die Bildschirmsteuerung im Textmodus benötigt. Mit 14,8 s Rechenzeit ohne Bildschirmausgabe ist der PC 1640 drei Sekunden langsamer als der Eazy-PC von Zenith (11,14 s) und etwa gleichschnell wie der M24 von Olivetti (15,10 s), der ebenfalls mit einem 8086 ausgerüstet ist. Mit der Bildschirmausgabe liegt der Schneider mit 49,8 s klar über dem Eazy-PC (33 s) und ist etwa um die Hälfte schneller als der Olivetti M24 (101 s). Dazu ist zu sagen, daß der M24 eine hochauflösende Mono-Grafik besitzt, die vom Standard abweicht, der Eazy-PC mit CGA-kompatibler Bildschirmelektronik ausgerüstet ist und der Schneider die EGA besitzt. In diesem Licht sind die Geschwindigkeiten der Bildschirmausgaben zu betrachten. Beim Schneider 35 s, beim Eazy-PC etwa 21 s, beim Olivetti 185 s. Allerdings: Benchmarks dürfen nicht überbewertet werden, denn bei PC-Kompatiblen zählen über die reine Geschwindigkeit hinaus die Kompatibilität in Hard- und Software und der Umfang der mitgelieferten Software.

Die Kompatibilität

Der Schneider 1640 besitzt nicht nur in Bezug auf die Hardware seine Eigenheiten. An manchen Stellen will er mit besonderer Software, mit seiner eigenen bedient werden. Zum Beispiel bedient er sich des Interrupts 15H, um die Bewegungen seiner Maus zu beobachten und seinen Uhrenbaustein mit nichtflüchtigem RAM zu bedienen. Das macht er zwar mit den Unterfunktionen, die in früheren PC-Generationen den Kassettenrekorder bedient haben (die deshalb von kaum einem anderen PC benutzt werden), aber es ist eine Eigenheit, die andere Kompatible nicht besitzen. Die Funktion der Maus und auch der Tastatur können außer Kraft gesetzt werden, wenn ein systemfremdes KEYBGR.COM benutzt wird. Letzlich melden sich Maus und Tasten in dem Puffer, aus dem Keybgr die Zeichen holt und an die interpretierende

Software weitergibt. Mir ging es mit systemfremden Keybgr so, daß nach Drücken der Taste Caps Lock alles stand und nur Abschalten half. Da wäre natürlich ein Reset-Knopf recht gewesen.
Der Schneider 1640 hat also einige kleine Eigenheiten. Diese stören aber im normalen Betrieb nicht. Alle Programme, die bei uns in der Redaktion Dienst tun, liefen anstandslos auf dem Rechner. Euroscript zum Beispiel, ein Textsystem, das Keybgr umgeht und selbst seine Umlaute und Sonderzeichen präpariert. Oder der Modula-Compiler aus dem System M2SDS von Interface Technologies, der mit Fenstern im EGA-Modus arbeitet. Die Programme, die Ulrich Cebulla geschrieben hat (mc 1987, Ausgaben 1, 2, 3, 4 und 9), um zu zeigen, wie man den Enhanced Graphic Adapter programmiert, liefen einwandfrei. Das heißt, daß die Implementation der EGA sehr kompatibel ist.
Die Schnittstellen des PC1640 arbeiten nach Standard. Es sind eine serielle und eine parallele Schnittstelle vorhanden. Beim Test früherer Modelle von Schneider wurde moniert, daß das Netzteil im Monitor des Gerätes eingebaut sei. Man könne es deshalb nicht auswechseln, wenn es zu schwach für die zusätzlich eingesteckten Karten geworden sei. Der PC1640HD20 ist aber in meinen Augen so komplett ausgerüstet, daß in den freien Slots nicht mehr große Erweiterungskarten eingesteckt werden müssen (weshalb auch die Tatsache, daß es nur drei Slots sind, anstelle der versprochenen vier, nicht weiter stört). Allerdings gilt das nur für Leute, die normale Bedürfnisse haben. Was geschieht, wenn man eine zusätzliche File-Card einstecken möchte, zusammen vielleicht mit Speicher nach der Erweiterungstechnik von Intel, Lotus, Microsoft, das sei dahingestellt, weil nicht ausprobiert. Es bleibt festzuhalten, daß ein normaler Anwender mit diesem Computer auf dem Schreibtisch sehr gut fährt, weil das Grundgehäuse geringere Ausmaße besitzt, als die Standardmodelle, die schon arg viel Platz wegnehmen.

Die Software

Der Schneider kommt mit einem umfangreichen Software-Paket. Mit GEM präsentiert er sich dem Benutzer. Der Schneider wird nicht nur in der EGA-Ausführung, sondern mit vielen Bildschirmvarianten geliefert. Das GEM paßt sich automatisch der vorhanden Variante in der Bildschirmdarstellung an. Beim EGA-PC1640 ist die Darstellung der Bildschirme recht präzise und das Arbeiten mit GEM macht Freude. In der Welt der Apple-Computer sind fast alle Programme an die Ikon-Menü-geführte Benutzeroberfläche angebunden und führen die Menüsteuerung intern analog fort. Das ist in der Welt der PCs mit GEM nicht so. Die meisten Programme sind ohne Rücksicht auf GEM entstanden und haben – wenn man überhaupt davon sprechen kann – eine eigene Benutzeroberfläche. Schneider liefert aber ein Basic mit, es heißt Basic2, das in GEM integriert ist und schöne Effekte besitzt. Für einen Menschen, der Basic knochentrocken in der Pionierzeit der Mikrocomputer gelernt hat, steckt es voller Überraschungen. Es kann mit numerierten Zeilen gearbeitet werden, muß aber nicht. Basic2 besitzt eine große Anzahl von Befehlen. Vor allem besitzt es Spezialbefehle, die den Umgang mit dem Bildschirm im Hinblick auf GEM erleichtern, die graphischen Fähigkeiten von Basic2 gut ausspielen und vieles in einem Schlag erledigen, was bei anderen Basics erst umständlich programmiert werden muß. Man befindet sich in einer eigenen Welt, mit allen Gefahren und Vorteilen. Die Gefahr: Man ist vom Rest der Welt abgekoppelt. Der Vorteil: Man kann viel mehr in Basic anstellen als andere, die nicht solch ein Basic besitzen.

Zitat des Monats

„Commondore ‚Amiga 500' ,..., Motorroller 68 000-Prozessor, 7 mHz Taktfrequenz mit Floppy-Laufwerken 3,5'".

Aus einem Anzeigenblättchen, gefunden von mc-Leser Herbert Demmel.

Ist er nun gut?

Die Tastatur hat bei meiner Schreibweise manchmal zu Buchstabenverdoppelungen geneigt. Den Anschlag benötigt der Schneider betont klar. Der Lüfter und die Festplatte machten ein deutlich hörbares Geräusch, das mir aber nicht unangenehm war. Für eingefleischte PCler mit Freude an den puren MS-DOS-Kommandos und komplexen Hochsprachenprogrammen erregt der Schneider-PC erst einmal den Verdacht, er sei nicht ganz kompatibel. Bei der Arbeit mit diesem Computer und der mitgelieferten Software kann man aber schnell herausfinden, daß der Computer in der Konfiguration im Test (ein Floppy-Laufwerk mit 360 KByte, 20-MByte-Festplatte, EGA-Karte, EGA-Monitor) ein hervorragend durchkonstruiertes Gerät ist, das für den Anwender im Büro perfekt geeignet ist. Dem technisch orientierten Anwender muß vorher klar sein, daß ein paar Probleme auftreten können, wenn es um den Ausbau in Bezug auf die Hardware geht. Da aber vieles schon eingebaut ist, betrifft das nur extreme Fälle. Im Versandhandel ist der Computer (ohne EGA und ohne Festplatte), ab ca. 1600 DM zu haben. In der getesteten Konfiguration muß man über 4000 DM anlegen. Nicht wenig, aber man erwirbt ein gut aussehendes und mit einem wirklich dicken Handbuch versehenes Gerät.

Das Handbuch

Das Handbuch des Schneiders enthält 706 beschriebene Seiten. Das ist wirklich viel, mancher wird vor der Dicke Respekt besitzen. Teil 1 schildert Grundsätzliches zum Umgang mit dem Computer von Amstrad und Schneider. Teil 2 behandelt das Arbeiten mit GEM Desktop. Teil 3 zeigt, wie man mit MS-DOS arbeitet. Teil 4 ist eine Einführung in Locomotive Basic2. Im Anhang stehen technische Details zum Computer. Da erfährt man, wie man Erweiterungskarten einsteckt, da sind die Fehlermeldungen von DOS gesammelt, und man kann auch nachlesen, wie die Systemuhr mit angeschlossenem batteriegepuffertem Zusatzspeicher programmiert wird. Dazu gibt es ein Programm namens NVR, das im Dialog mit Menüs die Einstellmöglichkeiten darstellt, Einstellungen annimmt und deren Abspeicherung im Non Volatible RAM vornimmt. Man kann den Bewegungsmaßstab der Maus einstellen, Tastenbelegungen verändern, die Größe der RAM-Disk einstellen (was manche Programme automatisch übernehmen) und die Standard-Bildschirmfarben einstellen.
Das Handbuch ist kompetent übersetzt, man findet kaum falsch übersetzte Fachbegriffe, wie es bei manchen anderen Handbüchern der Fall ist. Obwohl das Handbuch solche Dicke besitzt, ist es nicht als technisches Handbuch gedacht und kann es auch nicht sein. Der Platz reicht gerade, die drei großen Themen, das Arbeiten mit GEM Desk Top, mit MS-DOS und mit Basic2 zu schildern. Schneider ist in seinem Bemühen, dem Käufer hier viel Wissen an die Hand zu geben, recht weit gegangen. Dennoch muß man sich Zusatzliteratur besorgen, wenn man die Software kennenlernen will. Das liegt aber an der Komplexität des Hardware-Software-Systems und nicht an Schneider. Ein Schuß Kritik zum Schluß: Dieser schöne Computer besitzt leider in der EGA-Ausführung einen Bildschirm, der sehr spiegelt, was beim Arbeiten sehr stören kann, wenn man mit der Aufstellung nicht achtgibt. □

Peter Wollschlaeger

Devpac – Assembler der Superlative

Editor mit Assembler/Linker/Debugger für den Atari ST

Das Devpac-Paket von Hisoft wurde bisher als Geheimtip unter Atari ST-Assembler-Freaks gehandelt. Nachdem dieses Entwicklungspaket jetzt auch in Deutschland erhältlich ist, haben wir einen Insider gebeten, darüber zu berichten. Diese Software ist für Einsteiger und Profis interessant.

Es könnte der Eindruck entstehen, daß ich nicht ganz objektiv bin, denn ich habe das Handbuch zu diesem Assembler-Paket für den Atari ST übersetzt und auch sonst bei der „Eindeutschung" des Programmes mitgewirkt. Vorher habe ich allerdings zahlreiche Listings für ein ganzes Amiga-Buch mit dem fast gleichen Devpac-Assembler in der Amiga-Version geschrieben, und das hat natürlich seine Gründe. So ohne weiteres gebe ich ein gewohntes Tool nicht auf, doch zum Devpac-ST wechselte ich mit fliegenden Fahnen.

Geliefert wird neben dem Handbuch eine Diskette, auf der sich ein GEM-Bildschirm-Editor mit integriertem Makro-Assembler, zwei GEM-Debugger, zwei TOS-Debugger, ein Linker, ein Installationsprogramm für den Editor/Assembler und eine Datei mit den neuesten Informationen zum Devpac-ST befinden. Außerdem werden Quelltexte für ein Tutorial, eine komplette GEM-Applikation und zwei Include-Files für GEM geliefert.

Devpac selbst ist „uralt". Es wurde ursprünglich für den Sinclair QL geschrieben, dann (1985) an den Atari ST und neuerdings auch an den Amiga angepaßt. Die Atari ST-Version ist diejenige, die inzwischen am weitesten perfektioniert ist.

Der Kern des Devpac ist ein integrierter Editor/Assembler. Im Editor reicht ein Tastendruck, um ein ausführbares Programm zu erstellen. Auch kann linkbarer Code im GST-Format erzeugt werden, der dann mit dem Linker gebunden werden muß.

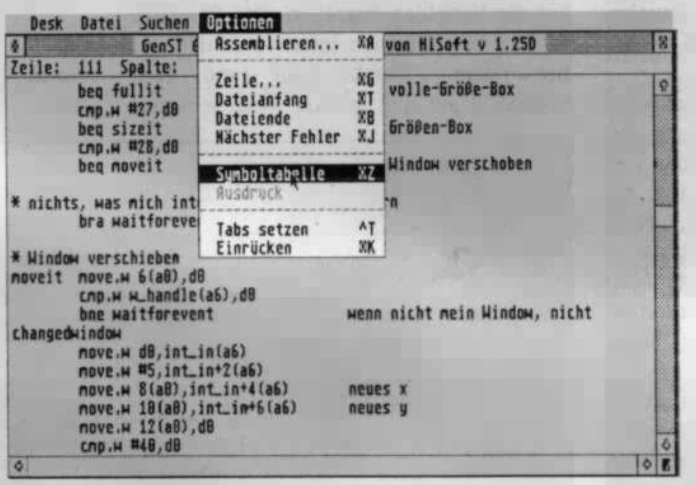


Bild 1. Der Editor des Devpac-ST ist auf das notwendigste beschränkt. Editor und Assembler sind ein Programm

GEM-Editor mit Wordstar-Tasten-Codes

Der Editor ist ein GEM-Editor mit den üblichen Fähigkeiten. Der Cursor kann wahlfrei mit den Tasten oder mit der Maus positioniert werden. Bewegt man sich mit den Cursor-Tasten durch einen längeren Text, muß man das lästige Nachlaufen beachten. Mit der Maus kommt man sicher besser und schneller zum Ziel. Die Bedienung des Editors kann über Pull-Down-Menüs erfolgen. Deren Items haben jedoch auch Tastatur-Kürzel, die über die ALT-Taste schneller eingegeben werden. Die Block-Kommandos „Definieren", „Sichern" und „Kopieren" werden über die Funktionstasten gesteuert. Der definierte Block wird nicht hervorgehoben. Ansonsten können Sie den GEM-Komfort vergessen und einige Wordstar-Kürzel (z.B. Control-Y zum Löschen einer Zeile) benutzen. Den Editor im Betrieb zeigt *Bild 1*. Er ist sicher nicht das Glanzstück dieses Software-Paketes, denn er kann nur einen Text in einem Fenster bearbeiten und bietet nur die wesentlichen Funktionen. Andere Editoren wie z. B. Tempus [1] sind wesentlich besser.

Extrem schnell

Der Editor/Assembler „GenST" enthält einen Zwei-Pass-Assembler, der nur vom Editor aus aufgerufen werden kann. Im ersten Durchlauf untersucht er den Text im Speicher und ggf. in Disketten-Dateien und baut daraus die Symboltabelle auf. Diese Tabelle ist vom Editor aus aufzulisten. Während des zweiten Durchlaufs erfolgt die Umsetzung der Befehle in 68000-Code. Eine Listing/Map-Datei kann dabei wahlweise auf dem Bildschirm, dem Drucker oder in einer Datei generiert werden. Die Option „nur Assemblieren" für einen Syntax-Test gibt es natürlich auch. Damit kann „auf die Schnelle" getestet werden, ob ein Programm richtig übersetzt wird. Aber auch wenn gleich die Programm- oder Objektcode-Datei geschrieben wird, ist er nur unwesentlich langsamer. Die Übersetzung erfolgt in einem atemberaubenden Tempo und kann jederzeit unterbrochen werden. Zum Beispiel brauchen 839 Zeilen Quellcode nur vier Sekunden. Danach steht bereits das ausführbare Programm auf der Festplatte. Mit einer RAM-Disk schafft man diese Übung in drei Sekunden. Das heißt, daß fast 17000 Zeilen in einer Minute assembliert und gebunden werden.

Im ersten Durchlauf werden nur fatale Fehler gemeldet (zum Beispiel Include-Datei nicht gefunden), die mit der Fehlermeldung „Tödlicher Fehler" sofort zum Abbruch des Assembler-Laufs führen. Während des zweiten Durchlaufes werden alle Fehler angezeigt, wenn gewünscht zusammen mit dem Listing und der Symboltabelle. Die

```
C+  Case-sensitiv (voreingestellt, wenn nicht anders installiert)
C-  Nicht case-sensitiv
D+  Debugger-Information → Binär-File
D-  Keine Debugger-Information → Binär-File (Voreinstellung)
L+  Erzeugt linkbaren Code
L-  Erzeugt ausführbaren Code (Voreinstellung)
M+  Expandiert Makros im Listing
M-  Keine Makro-Expansion im Listing (Voreinstellung)
N+  Schmales Listing
N-  Breites Listing (Voreinstellung)
S+  Symboltabelle mit Listing (durch LIST gesetzt)
S-  Keine Symboltabelle (durch NOLIST gesetzt)
W+  Warnmeldungen ausgeben
W-  Warnmeldungen unterdrücken
```

Bild 2. Mit diesen Direktiven wird der Assembler gesteuert

Fehlermeldungen werden intern gespeichert. Im Editor kann man dann die fehlerhaften Zeilen nacheinander anfahren (Funktion „Gehe zu Fehler") und verbessern. Der Assembler-Lauf kann mit der OPT-Direktive, gefolgt von einem oder mehreren Kürzeln in Unix-Schreibweise, gesteuert werden. Die zur Auswahl stehenden Direktiven zeigt *Bild 2.*

Der Assembler verkraftet klaglos auch komplexe Ausdrücke. Die zur Verfügung stehenden Operatoren, in absteigender Reihenfolge der Priorität aufgeführt, sind: minus (-) und plus (+) als Vorzeichen, bitweise NOT (~), links (<<) und rechts (>>) schieben, Gleichheit (=), bitweises AND (&), OR (!) und XOR (↑), Multiplikation (*) und Division (/), Addition (+) und Subtraktion (-).

Die Priorität kann durch Klammern aufgehoben werden. Ausdrücke mit Operatoren gleicher Priorität werden von links nach rechts entwickelt. Alle Ausdrücke werden in 32-Bit-Arithmetik mit Vorzeichen gerechnet, allerdings ohne Test auf Überlauf.

Spezielle Direktiven für Daten-Strukturen

Gerade bei GEM-Programmen benötigt man immer Offset-Tabellen. Diese mühsam über EQUates zu erzeugen, ist natürlich auch möglich, aber besser macht sich „RS" mit Zubehör. Die RS-Direktive erlaubt, eine Liste von Konstanten-Definitionen zu erstellen, was sehr nützlich bei Datenstrukturen und globalen Variablen ist. Am besten zeigt man die Funktion an einem Beispiel. Angenommen, Sie greifen auf alle Variablen als Offset von A6 zu (wie das beim Editor/Assembler und den Debuggern geschieht), so können Sie sie definieren mit Zeilen wie diesen:

```
onstate  rs.b 1
start    rs.l 1
end      rs.l 1
```

Sie können dann (nachdem A6 geladen wurde) in der Form

```
move.b   onstate(a6),d1
move.l   start(a6),d0
cmp.l    end(a6),d0
```

auf die Variablen zugreifen. Die Direktive RS hat ihren eigenen internen Zähler, der bei jedem Assembler-Lauf (und nach RSRESET) auf null gesetzt wird. Jedesmal wenn der Assembler diese Direktive bearbeitet, weist er dem Label den aktuellen Wert dieses Zählers zu (mit Wortjustierung bei .W und .L) und inkrementiert dann den Zähler entsprechend dem Typ (.B, .W, .L) und der Größe von „Ausdruck". Im obigen Beispiel wird „onstate" den Wert 0 haben, „start" den Wert 2 (wurde auf Wortgrenze justiert) und „end" den Wert 6.

Als weiteres Beispiel lassen Sie uns annehmen, Sie haben eine Datenstruktur, die aus einem Langwort, einem Byte und noch einem Langwort besteht, und zwar in dieser Reihenfolge. Um Ihr Programm möglichst gut lesbar und leicht änderbar zu gestalten, könnten Sie Zeilen wie die folgende einsetzen:

```
        rsreset
d_next  rs.l        1
d_flag  rs.b        1
d-where rs.l        1
```

Dann können Sie darauf zugreifen, wie

```
move.l      d_next(a0),a1
move.l      d_where(a0),a2
tst.b       d_flag(a0)
```

```
* Source code Copyright (C) 1985 HiSoft. All rights reserved.

* a macro to initialise a program
do_start MACRO
        move.l a7,a5
        lea datastart(pc),a6              data pointer
        lea datalength+4000(a6),a7        my stack pointer
        move.l 4(a5),a5                   base page
        move.l $c(a5),d0                  text
        add.l $14(a5),d0                  data
        add.l $1c(a5),d0                  bss
        add.l #$100+datalength+4000,d0    base page length+data space
* firstly we have to give back the memory not required
        move.l d0,-(a7)                   length to keep
        move.l a5,-(a7)                   start address
        clr.w -(a7)                       filler
        move.w #$4a,-(a7)
        trap #1                           M_SHRINK
        lea 12(a7),a7
        ENDM

* a macro to call a VDI function given as a parameter
vdi     macro
        move.w #\1,d0             cant be Quick as some are>127
        bsr call_vdi
        endm

* a macro to call GEM function given as a parameter
gem     macro
        moveq #\1,d0              can be Quick as all are<128
        bsr call_gem
        endm

* initialise the VDI parameter block
initvdipars
        lea vdipar(a6),a1
        lea contrl(a6),a0
        move.l a0,(a1)+
        lea intin(a6),a0
        move.l a0,(a1)+
        lea ptsin(a6),a0
        move.l a0,(a1)+
        lea intout(a6),a0
        move.l a0,(a1)+
        lea ptsout(a6),a0
        move.l a0,(a1)+
        rts
```

Bild 3. Ein Auszug aus der Include-Datei zur GEM-Programmierung. Beachten Sie, wie am Ende des Listings die Arrays festgelegt werden

```
* call the vdi, with parameter in d0
* all other parameters have to be set by caller
call_vdi
        move.w d0,contrl(a6)            store the op-code
        lea vdipar(a6),a0
        move.l a0,d1                    the parameter block
        moveq #115,d0                   the function code
        trap #2
        rts


* initialise the GEM array params[0..5]
initgempars
        lea params(a6),a1
        lea control(a6),a0
        move.l a0,(a1)+
        lea global(a6),a0
        clr.l 10(a0)                    ap_ptree
        clr.l 14(a0)
        clr.l 18(a0)
        clr.l 22(a0)
        clr.l 26(a0)
        move.l a0,(a1)+
        lea int_in(a6),a0
        move.l a0,(a1)+
        lea int_out(a6),a0
        move.l a0,(a1)+
        lea addr_in(a6),a0
        move.l a0,(a1)+
        lea addr_out(a6),a0
        move.l a0,(a1)+
        rts


* call GEM function D0.W - all elements in control[1..4] are set
call_gem
        move.w d0,control(a6)           store the op code
        sub.w #10,d0
        mulu #3,d0                      multiply by 3
        lea gem_ctrl_list(pc),a0
        add.w d0,a0                     points to the entry
        moveq #0,d0
        move.b (a0)+,d0
        move.w d0,control+2(a6)         do control(1)
        move.b (a0)+,d0
        move.w d0,control+4(a6)         and control(2)
        move.b (a0)+,d0
        move.w d0,control+6(a6)         and control(3)
        clr.w control+8(a6)             control(4)=0
        cmp.w #RSRC_GADDR,control(a6)
        bne.s c6is0
* this is a special case which has control(4) as 1
        move.w #1,control+8(a6)
c6is0   lea params(a6),a0
        move.l a0,d1                    the pointer
        move.w #200,d0                  function number
        trap #2
        move.w int_out(a6),d0   usually a returned value
        rts

*****************************
*       WORKSPACE           *
* always addressed off A6 *
*****************************

* this is the array of pointers to the rest of the AES arrays
params          rs.l 6

* these are the arrays for AES
control         rs.w 5
global          rs.b 30
int_in          rs.w 16
int_out         rs.w 7
addr_in         rs.l 2
addr_out        rs.l 1


* these are the VDI arrays, starting with the parameter block
vdipar          rs.l 5

* now the arrays themselves
contrl          rs.w 12
intin           rs.w 30
ptsin           rs.w 30
intout          rs.w 45
ptsout          rs.w 12
```

Die RSRESET-Direktive setzt den internen Zähler der RS-Direktive zurück (Reset auf 0). Natürlich lassen sich so auch Strukturen erzeugen, bzw. die Offset-Tabellen dafür. Über deren Platzbedarf muß man nun nicht einmal nachdenken.

Die RSBSS-Direktive setzt die Größe der BSS-Sektion des Programms auf den aktuellen Wert des RS-Zählers. Wie elegant man auf diese Art das leidige Problem der AES- und VDI-Arrays lösen kann, sehen Sie am Ende des Programmes in Bild 3. Das Listing ist ein Auszug aus dem mitgelieferten Include-File. Aus Copyright-Gründen wurde es so gekürzt, daß es nicht lauffähig ist.

Es gibt im Devpac-Paket viele Direktiven, die zum bedingten Assemblieren eingesetzt werden. Zu Beginn eines „bedingten" Blocks muß eine der vielen IF-Direktiven stehen, am Ende des Blocks ein ENDC. Die „bedingten" Blöcke können bis zu 32 Ebenen tief geschachtelt werden. An IF-Direktiven werden geboten: IFEQ, IFNE, IFGT, IFGE, IFLT, IFLE (Bild 4).

Diese Direktiven entwickeln den jeweils angegebenen Ausdruck, vergleichen seinen Wert mit null und schalten in Abhängigkeit vom Ergebnis das bedingte Assemblieren an oder aus. Die Bedingungen selbst korrespondieren exakt mit den Bedingungs-Codes der CPU 68000. Wenn z.B. das Label DEBUG den Wert 1 hat, wirken die folgenden Anweisungen

```
IFEQ DEBUG
logon dc.b 'Enter a command: ',0
ENDC
IFNE DEBUG
opt d+
logon dc.b 'You man, gimme: ',0
ENDC
```

Sehr leistungsfähige Makros

Der Assembler ist ein Makro-Assembler, der sich zuerst an den Motorola-Standard hält, dann aber durch Leistung glänzt. Was damit gemacht werden kann, soll am Beispiel von n! (Sie lesen richtig: Fakultät) gezeigt werden. Ganz praxisgerecht ist das nicht, es zeigt aber sehr schön, wie Rekursionen in Makros funktionieren können. Bevor das Makro gezeigt wird, ist es nützlich, sich anzusehen, wie das in einer Hochsprache wie Pascal gelöst wird:

```
function factor(n:integer):integer;
  begin
   if n>0 then
     factor:=n*(factor-1)
    else
   factor:=1
   end;
```

Die Makro-Definition nutzt die SET-Direktive für Multiplikationen wie n*(n-1)*(n-2) usw. auf diese Art:

```
factor MACRO
       IFND   \1
\1     set    1        (setze 1 für
                       ersten Aufruf)
       ENDC

       IFGT   \2
       factor \1,\2-1  (nächste Ebene
                       abarbeiten)
\1     set    \1*(\2)  (n=n*(factor-1)
       ENDC

       ENDM
```

Diese Funktion wird beispielsweise so aufgerufen:
factor test,3

Das nette Ergebnis des obigen Quelltextes ist, daß die Variable Test auf 3! (Fakultät von 3) gesetzt wird. Der Grund dafür, daß beim zweiten SET (\2) anstatt \2 geschrieben wurde, ist, daß dieser Parameter normalerweise nicht ein simpler Ausruck ist, sondern eine Liste von Zahlen, durch Minuszeichen getrennt. Das könnte also so assembliert werden:

test set test*5-1-1-1

d. h. test*5-3 anstatt des richtigen

set test*(5-1-1-1)

Linker-Direktiven im Quelltext

Wenn die Direktive OPT L+ genutzt oder die Voreinstellung mit dem Installationsprogramm geändert wurde, produziert der Assembler linkbaren anstatt ausführbaren Code. In diesem Modus sind die folgenden Direktiven zusätzlich verfügbar:

„MODULE string" setzt den Namen „string" als Modulname ein. „SECTION string" gibt den SECTION-Namen für eine Datei vor. „XDEF label <,label,label u.s.w>" definiert Label(s) als „public", d.h. als sichtbar für andere Dateien, die zu dieser Datei gelinkt werden. „XREF label <,label,label u.s.w>" definiert die Label(s) als externe Referenzen, d.h. sie werden erst beim Linkerlauf importiert und sind beim Assemblerlauf unbekannt. Mit XREF werden relative Labels, d.h. Programmadressen definiert, wohingegen mit XREF.L absolute Labels, d.h. Konstanten definiert werden. Eine Liste aller Assembler-Direktiven zeigt *Bild 4.*

Motorola-Standard

Der 68000-Befehlssatz wird voll unterstützt, einige Abkürzungen werden akzeptiert und automatisch umgesetzt. Der Assembler konvertiert automatisch in die A- und I-Form, wenn erforderlich, allerdings bei einigen Befehlen nicht in die M- oder Q-Form. Sie können aber auch selbst das A oder I bei den Befehlen anfügen. Zum Standard-Assembler gibt es nur einen Unterschied. Das GEMDOS-Programm-Modell für den Atari ST schreibt vor, daß jede ausführbare Datei drei Sektionen (Text, Data und BSS) haben sollte. Der Assembler von Devpac-ST selbst unterstützt dieses Modell nicht vollständig. Der erzeugte Code (Daten sind auch Code) wird im Text-Segment abgelegt, das Data-Segment ist immer leer.
Die BSS-Sektion bleibt leer, solange nicht RSBSS und DSBSS angezogen werden. Auf diese vornehme englische Art versucht Hisoft auszudrücken, daß es Nonsens ist, zwischen Text und Data zu unterscheiden, wenn der Lader doch immer beide Segmente direkt hintereinander im Speicher ablegt.

END	Ende Quelltext
INCLUDE	Liest Quelltext von Disk
OPT	Optionen folgen
EVEN	PC auf gerade Adresse
CNOP	Justiert PC
ORG	Startadresse für absoluten Code
DC	Definiere Konstante(n)
DS	Definiere Speicher
DSBSS	Speicher in BSS reservieren
FAIL	Erzwinge Fehlermeldung
LIST	Listing an
NOLIST	Listing aus
LLEN	Seitenlänge setzen
PLEN	Zeilenbreite setzen
TTL	Titel setzen
SPC	Leerzeilen ausgeben
PAGE	Seitenvorschub
LISTCHAR	Kontrollzeichen ausgeben
EQU	Wert Label setzen
EQUR	Register-Equate
SET	Temporären Wert einer Label setzen
RS	Reserviere Speicher
RSRESET	Reset RS-Zähler
RSBSS	Größe der BSS setzen
IFEQ	null
IFNE	nicht null
IFD	wenn Label definiert
IFND	wenn Label nicht definiert
IFGT	wenn größer als
IFGE	wenn größer oder gleich
IFLT	wenn kleiner als
IFLE	wenn kleiner oder gleich
IFC	wenn Strings gleich
IFNC	wenn Strings nicht gleich
ENDC	Ende bedingtes Assemblieren
MACRO	Start der Definition
ENDM	Ende der Definition
MEXT	Erzwingt Ende Makroexpansion
MODULE	Setze Modul-Name
SECTION	Sections-Name
XDEF	Definiert interne Labels als public
XREF	definiert externe, zu importierende Labels

Bild 4. Alle Compiler-Direktiven im Überblick

Mächtige Debugger

Zum Devpac-Paket gehören vier Debugger. Es handelt sich dabei jeweils um symbolische Debugger/Disassembler. Diese Vielfalt hat zwei Gründe. Zuerst gibt es je eine GEM- und eine TOS-Version. Damit werden die Konflikte zwischen dem TOS-Cursor und der Maus vermieden. Zum zweiten gibt es noch von beiden Debuggern die A-Versionen, was für autoresident steht. Diese Versionen sollten im AUTO-Ordner der Boot-Diskette installiert werden, damit sie von Anfang der Arbeit an im System schlummern. In einem Atari ST sollten immer die 14 KByte für den Debugger plus 32 KByte für den seperaten Video-RAM zur Verfügung stehen.
Nun können Sie ohne Risiko Ihre selbstgeschriebenen Programme laufen lassen. Sollte es abstürzen (welches neue Programm tut das nicht?), sehen Sie nun keine Bomben, sondern landen automatisch im Debugger. Was die Debugger mit den Exception-Vektoren machen, zeigt *Bild 5.*
Bei den nicht residenten Versionen müssen Sie erst den Debugger und danach Ihr Programm laden. Die „Entwanzer" benutzen nur sehr wenige TOS-Routinen. Damit es noch funktioniert, sollten die folgenden Funktionen intakt sein:

```
BIOS 1 bconstat von KBD
BIOS 2 bconin von KBD
BIOS 3 bconout zu AUX und PRT
XBIOS 0 initmouse
XBIOS 33 setprt
XBIOS 38 supexec
```

Für die Video-Ausgabe haben die Debugger einen eigenen, sehr schnellen Video-Treiber. Für das zu testende Programm gibt es einen eigenen Video-RAM (daher die extra 32 KByte). Man kann jederzeit zwischen dem Debugger-Display *(Bild 6)* und dem Programm-Output umschalten. Das Display selbst ist auch noch umschaltbar, um ganzseitig Speicherauszüge oder Disassembler-Listings darstellen zu können. Der Debugger ist trotz seiner nur 14 KByte Umfang sehr leistungsfähig und sehr schnell. Alle erlaubten Befehle sind in *Bild 7* zusammengestellt.

Wertvolle Anhänge

In den Anhängen des Handbuches findet man eine Fülle wertvoller Informationen. File-Header-Format, Base-Page oder Speicherbelegung sind nur ein paar Stichworte. Dazu kommt eine komplette Liste aller GEMDOS-, BIOS- und XBIOS-Funktionen und eine Beschreibung über das Einbinden von Maschinensprache-Routinen in ST-Basic. Sehr gut gemacht ist der Abschnitt über den Aufruf von System-Routinen.
Wie gefällt Ihnen z. B. das: Die AES-Rechtecke werden in der Form „links, oben, Breite, Höhe" angegeben, die VDI-Rechtekke hingegen als „links, oben, rechts, unten". In Assembler ist die Konvertierung sehr einfach, besonders wenn Sie Datenregister und den MOVEM-Befehl einsetzen. Zum Beispiel kann das AES-Rechteck mit intout[1-4] in das VDI- Format mit ptsin[0-3] so umgerechnet werden:

```
movem.w  intout+2(a6),d0-d3
add.w    d0,d2              x rechts
add.w    d1,d3              y unten
movem.w  d0-d3,ptsin(a6)
```

Versuchen Sie einmal, das in C effizienter zu lösen!
So nette kleine Tricks sind übrigens des öfteren im Handbuch eingestreut. Wenn man so programmiert, ist auch erklärlich, daß der GEM-Editor mit dem Assembler und integriertem Linker nur 30 KByte (plus 4 KByte RSC-Datei) belegen. Das erklärt auch das hohe Tempo. Über die (sehr gute) Qualität des Handbuchs von rund 160 Seiten will ich mich nicht äußern, weil ich es übersetzt habe. Auf jeden Fall ist es so geschrieben, daß auch der Einsteiger in die Assembler-Programmierung damit sehr gut zurecht kommen dürfte.

Literatur

[1] *Schönrock, Reiner:* Editor für Profis, mc 11/87, Seite 137

Exc.-Nr.	Bezeichnung	Normaler Effekt	Effekt unter MonST
2	Bus-Fehler	Bomben	Abgefangen
3	Adreß-Fehler	Bomben	Abgefangen
4	Illegaler Befehl	Bomben	Abgefangen
5	0-Division	Bomben	Abgefangen
6	CHK-Befehl	Bomben	Abgefangen
7	TRAPV	Bomben	Abgefangen
8	Privilegverletzung	Bomben	Abgefangen
9	Trace	Bomben	Einzelschritt
10	Line-A-Emulator	Line-A-Grafik	Line-A-Grafik
11	Line-F-Emulator	TOS intern	TOS intern
32	Trap #0	Bomben	Abgefangen
33	Trap #1	GEMDOS-Aufruf	GEMDOS-Aufruf
34	Trap #2	GEM-Aufruf	GEM-Aufruf
35–44	Trap #3–#12	Bomben	Abgefangen
45	Trap #13	BIOS-Aufruf	BIOS-Aufruf
46	Trap #14	XBIOS-Aufruf	XBIOS-Aufruf
47	Trap #15	Bomben	Abgefangen

Bild 5. So entschärfen die vier Debugger des Devpac-ST die Atari ST-Bomben

```
D0:00000007 20 0001 B1CA 0001 B1  A0:0001DDB2 0000 0000 6000 0000
D1:00006000 0100 0000 0000 0000   A1:00026AF6)6000 021A 43EE 007A
D2:00000000 601E 0100 00FC 0020   A2:00026A78 0000 0000 0000 0000
D3:000269F6 0002 69F6 000F 8000   A3:00000000 601E 0100 00FC 0020
D4:0000FFFF 28 3F39 0001 2F60 3F  A4:000F7FF8 0000 0000 0002 69F6
D5:00000000 601E 0100 00FC 0020   A5:0001AD20 0001 AD20 000F 8000
D6:00000000 601E 0100 00FC 0020   A6:0001DD04 0000 0000 0000 0000
D7:00000006 0020 0001 B1CA 0001   A7:000F7FF8 0000 0000 0002 69F6
PC:00026AF6 SR:0308  U                       A7'00004DB8 0100 0100 0280 0000
```

```
00026AE6 0000 0000 0000 0000 0000 0000 0000 0000  ................
00026AF6)6000 021A 43EE 007A 41EE 008E 22C8 41EE  `...C..zA...".A.
00026B06 00A6 22C8 41EE 00E2 22C8 41EE 011E 22C8  ..".A..."."A..."
00026B16 41EE 0178 22C8 4E75 3D40 008E 41EE 007A  A..x".Nu=@..A..z
```

```
00026AF6          )BRA MAIN
00026AFA INITVDIP LEA $007A(A6),A1
00026AFE          LEA $008E(A6),A0
00026B02          MOVE.L A0,(A1)+
```

```
Befehl:
```

GEM MonST Disassembler/Debugger 1.25D (C) HiSoft 1986 - Autor A.Pennell

Bild 6. Das Standard-Display der Debugger, die einen separaten Video-RAM haben

G	Get (Suche) Folge	Ctrl-A	Breakpoint nach aktuellem Befehl, Befehl ausführen
H	Hilfe, zeigt Base-Page-Variable	Ctrl-B	Breakpoint setzen/löschen
I	Intelligente Kopie	Ctrl-C	MonST verlassen
L	Labels listen	Ctrl-K	Kill (lösche) alle Breakpoints
M	Modifiziere Speicherzeiger	Ctrl-R	Resumee (Fortsetzen) nach Exception
N	Nächstes finden (nach G)	Ctrl-T	Unterprogramm (und Trap) ausführen
O	Other (andere) Zahlenbasis zeigen	Ctrl-X	Speicher auf Drucker (oder File) ausgeben
P	Print Disassembler-Listing auf Drucker oder Schirm	Ctrl-Y	Einzelschritt
Q	Quick (schnelles) Disassemblieren		
R	Register-Wert ändern	0–9, A–F	Hex-Eingabe in Speicher
T	Temporär Speicherzeiger ändern	"	ASCII-Eingabe in Speicher
U	go until (bis)	[Return]	Speicherzeiger um 1 Wort inkrementieren
V	Video umschalten	–	Speicherzeiger um 1 Wort dekrementieren
W	fülle Speicher mit (with)	[Tab]	Speicherzeiger auf Wert von PC
X	eXaminiere Speicher	Cursor-Tasten: Speicherzeiger verschieden ändern	

Bild 7. Die Kommandos der Debugger

Jürgen Plate

Turbo-Pascal wird professionell

Turbo-Pascal Version 4

Nach Turbo-Basic und Turbo-C war es eigentlich zu erwarten, daß auch der gute alte Pascal-Compiler aus Borlands Softwareschmiede ein neues Gewand bekommt. Nun ist er da, und es sind nicht nur die Kleider neu. Vor allem das Unit-Konzept macht ihn für professionelle Anwendungen interessant.

Neu ist – ganz offensichtlich für den Anwender – die Benutzeroberfläche, die nun genauso aussieht wie bei den anderen Compilern von Borland. Das Fenster- und Menükonzept ist fast so wie bei Turbo-C (*Bild 1*). Der Umsteiger von der alten Turbo-Version muß sich etwas umgewöhnen, hat aber dafür viele Vorteile: Online-Hilfsfunktionen, Dateiauswahl per Cursor, ein echter Ausgang ins Betriebssystem (man kann sogar in der Bildschirmausgabe noch nachsehen, wenn man schon längst wieder zurück im Compiler ist), die aktuelle Datei läßt sich unter einem anderen Namen speichern und vieles mehr. Der Editor ist bis auf ein paar marginale Änderungen geblieben wie er war. Besonders angenehm ist es, sich per „Pick" eine von den zuletzt bearbeiteten Dateien „herauspicken", eine sehr bequeme Sache, wenn man zwischen verschiedenen Include-Dateien pendelt. Die Pick-Namen kann man sogar – genauso wie die gesamte Systemkonfiguration – auch auf Diskette speichern. Beim nächsten Aufruf von Turbo landet man dann automatisch in der zuletzt bearbeiteten Quelle.

Überhaupt läßt sich die Compilation jetzt komplett von der Quell-Datei aus steuern. Alles, was man bei Turbo-3 über das Optionenmenü (Compiler Options) einstellen mußte, kann man jetzt in der Quell-Datei eintragen, und somit auch nicht mehr vergessen. Deshalb ist es wie bei Turbo-C jetzt mit der Kommandozeilenversion des Compilers möglich, Programme per Batch-Auftrag zu übersetzen. Wie bei C gibt es auch eine Build/Make-Option, die das komplette Nachübersetzen ganzer Projekte erlaubt. Sie sehen schon, Turbo-Pascal ist zu einem richtigen Werkzeug für Entwickler geworden. Man merkt es auch am Compiler; früher kam er mit 39 KByte aus, jetzt braucht er 115 KByte. Daß davon nicht alles nur für den Benutzerkomfort draufgeht, zeigt die Kommandozeilenversion mit 41 KByte. Nun aber zu dem, was sich unter der schönen Oberfläche getan hat:

- Aufhebung der 64-KByte-Grenze (EXE-Dateien)
- Intelligenter Linker (Einbinden von OBJ-Dateien)
- DOS- und Grafikunterstützung
- Erzeugen vorübersetzter Module (Units)

Der Compiler erzeugt nun EXE-Dateien, der Code kann also den gesamten freien Speicherplatz belegen. Dabei wird er in sogenannte Units unterteilt, von denen jede kleiner als 64 KByte sein muß. Auf die Units komme ich weiter unten noch zu sprechen. Die Overlays sind leider ersatzlos gestrichen worden, was mich persönlich veranlaßt, einige Programme, die sehr viel Speicherplatz für Daten brauchen, weiter mit Turbo-3 zu fahren. Eine weitere wichtige Neuerung ist sicher die Unterstützung des Arithmetikprozessors (8087, 80287, 80387), die Generierung von MAP-Dateien für Laufzeit-Debugger (z. B. SYMDEB) und die bedingte Übersetzung, die Real-Arithmetik nach ANSI-Norm (wie bei C) und der neue Linker. Es werden bei der Code-Erzeugung nur noch die wirklich benötigten Teile der Laufzeitbibliothek hinzugebunden, was die erzeugten Code-Dateien bei kurzen Utilities stark schrumpfen läßt. So braucht ein leeres Programm (BEGIN-END.) nur noch knapp über 1 KByte Code.

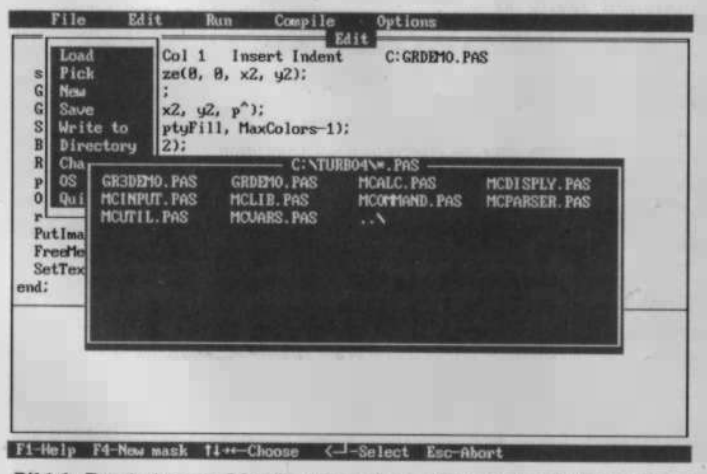


Bild 1. Das bekannte Menüsystem gibt es jetzt auch bei Turbo-Pascal

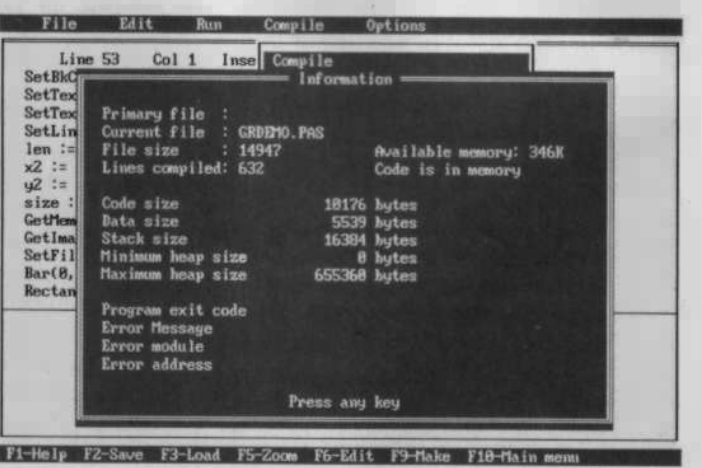


Bild 2. Die Compilerstatistik gibt Aufschluß über das Programm

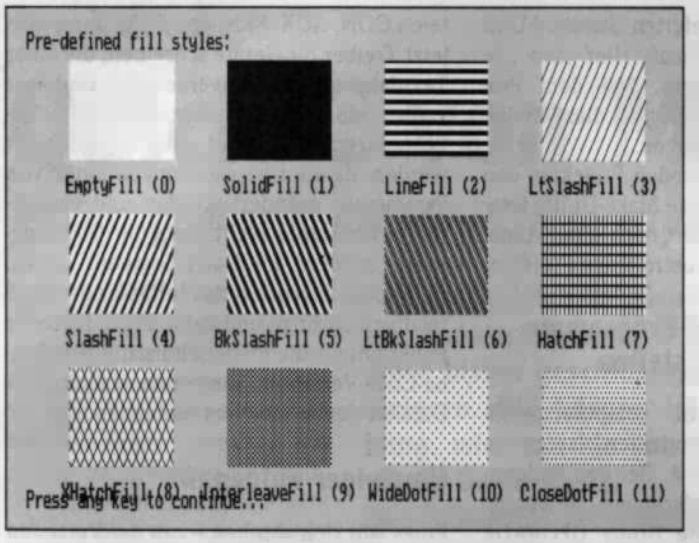


Bild 3.
Die vordefinierten Schraffurmuster der Grafikroutinen

```
Unit StackLib;

Interface

  Function  Empty : Boolean;
  Procedure Push (X: Integer);
  Procedure Pull (var X: Integer);

Implementation

Type   Stack_Ptr = ^I_Stack;
       I_Stack = Record
                   Wert : Integer;
                   Next : Stack_Ptr;
                 End;
  Var Root         : Stack_Ptr;

  Procedure Push (X : Integer);
    Var P : Stack_Ptr;
    Begin
    New (P);
    with P^ do
      begin Wert := X; Next := Root end;
    Root := P;
    End;

  Procedure Pull (var X : Integer);
    Var P : Stack_Ptr;
    Begin
    P := Root;
    with P^ do
      begin X := Wert; Root := Next end;
    Dispose (P);
    End;

Function Empty : Boolean;
  begin Empty := Root = nil end;
begin
  Root    := nil;
end.
```

Bild 4.
Ein einfaches Beispiel für eine UNIT

Mit Turbo-4 ist auch endlich die Zeit der hexadezimalen Eingabe von Inline-Zeilen vorbei: der Linker kann OBJ-Dateien, die mit einem Assembler (z. B. MASM) erzeugt werden, direkt in den Code einbinden. *Bild 2* zeigt die Statistik nach der Übersetzung eines Programms.

Neue Typen und Funktionen

Turbo-Pascal 4.0 hat auch bei vordefinierten Typen und Funktionen Neues zu bieten. Der Typ INTEGER ist nun auf Werte zwischen −32768 und +32767 begrenzt. Neu bei den ganzen Zahlen sind die Typen SHORTINT (−128...127), Word (0...$FFFF) und LONGINT (−2147483648 bis 2147483647). BYTE gibt es natürlich weiterhin. Bei den Gleitpunktzahlen sind die Typen eingeteilt in

REAL (1E-39...1E38, 11 Stellen),
SINGLE (1E-45...1E38, 8 Stellen),
DOUBLE (1E-324...1E308, 15 Stellen),
EXTENDED (1E-4951...1E4932,
19 Stellen) und
COMP (−2 ↑ 63+1..2 ↑ 63−1, 8 Byte).

Bei den Prozeduren und Funktionen ist eine ganze Palette von DOS- und Grafikfunktionen hinzugekommen. So kann man beispielsweise nun statt I := I + 7 jetzt INC(I,7) schreiben. Analoges gilt für die DEC-Funktion. Viele Betriebssystemfunktionen sind nun ohne eigenen Programmieraufwand erreichbar. Es gibt Prozeduren für Datum und Uhrzeit, zur Interrupt-Behandlung, zum Zugriff auf das Inhaltsverzeichnis, alle DOS-Devices sind direkt von Turbo aus erreichbar und IORESULT liefert die Original-Fehlernummern des Betriebssystems. Mit der EXEC-Funktion ruft man Kind-Prozesse auf. Der Zugriff auf den Bildschirm ist schneller geworden – auch das direkte Schreiben in den Bildschirmspeicher ist möglich. Für den Zugriff auf die Tastatur wurde „READ(KBD,CH)" durch „CH := READKEY" ersetzt. Beim Abfragen der Tastatur wird bei Funktions- und Sondertasten nicht mehr der Escape-Code (27), sondern wie beim DOS eine Null als erstes von zwei Zeichen geliefert. APPEND ist nun auch bei Textdateien möglich. Dynamische Variablen werden in ihrer wirklichen Größe auf dem Heap abgelegt – der jetzt byteweise verwaltet wird. Neu ist auch der generische Typ POINTER, der zu jedem Pointertyp verträglich ist. Man benötigt ihn für die weiter unten beschriebenen Units.

Die Grafikroutinen unterstützen eine ganze Reihe von Adapterkarten: CGA, EGA, Hercules, VGA und andere. Bei der Vorab-Version des Compilers war noch nicht alles vollständig implementiert. *Bild 3* gibt einen kleinen Eindruck von der Mächtigkeit der Grafikroutinen – es stellt die vordefinierten Muster zum Füllen von Polygonen dar. Insgesamt gibt es sechs Units in der Standard-Bibliothek:

- SYSTEM mit dem Standard-Prozeduren und-Funktionen
- CRT mit den aus Turbo-3 bekannten Sound- und Bildschirmroutinen
- PRINTER für die Druckersteuerung
- GRAPH mit zahlreichen Grafikroutinen
- GRAPH3 und TURBO3 zur Unterstützung bei der Umstellung

Das Unit-Konzept

Dies ist die wohl umfassenste und gleichzeitig interessanteste Erweiterung gegenüber der alten Turbo-Version, da hier das Unit- Konzept des ehrwürdigen UCSD-Pascals wiederauferstehen, aber auch das richtungsweisende Modulkonzept von Modula 2 anklingt. Mit den Units werden zwei Vorteile in Turbo-Pascal eingebracht:

– Da jede Unit bis zu 64 KByte groß sein darf, ist das Gesamtprogramm nur noch vom verfügbaren Speicher begrenzt.

– Die Unterteilung des Programms in Units unterstützt zudem die Entwicklung großer Programme und die Erstellung und Verwaltung von Standardbibliotheken. Das ist besonders dann wichtig, wenn mehrere Leute an einem Softwareprojekt arbeiten.

Eine Unit ist ein vollständig abgeschlossener Programmteil mit definierter Schnittstelle, der getrennt übersetzt werden kann. Das Schema einer Unit ist in *Bild 4* gezeigt. Statt des Schlüsselworts „PROGRAM" beginnt sie mit „UNIT". Die Schnittstelle zu

anderen Programmteilen oder Units wird mit „INTERFACE" eingeleitet. Hier werden die Deklarationen aufgeführt, die nach außen sichtbar sein sollen, also von anderen Programmteilen verwendet werden können. Bei Modula heißt dieser Teil „Definition Module". Bei Prozeduren und Funktionen wird im Interface-Teil nur der Prozedurkopf einschließlich Parameterliste aufgeführt. Der anschließende, durch „IMPLEMENTATION" eingeleitete Teil enthält alle Deklarationen, die nicht nach außen wirken sollen und die im Interface-Teil definierten Routinen mit ihren Anweisungen. Es schließt sich wie im Hauptprogramm ein Anweisungsteil (BEGIN...END.) an, in dem Werte initialisiert werden können. Diese Anweisungen werden vor dem Start des eigentlichen Programms abgearbeitet. Am Beispiel eines einfachen Stacks in (Bild 4) sieht man das sehr deutlich. Die Units werden auf Dateien mit der Endung TPU abgelegt. Das Einbinden der Units erfolgt in anderen Programmen durch „USES Datei" (der Name hinter UNIT wird nicht ausgewertet). Um Units allgemeiner verwendbar zu machen, wurde der Typ POINTER eingeführt, der zu jeder Zeigervariablen verträglich ist. Beim Aufruf einer Unit wird zunächst in der mitgelieferten Standard-Unit TURBO.TPL nachgeschaut. Hier sind alle Standard-Units enthalten. Mit dem Programm TPUMOVER kann man diese Bibliothek erweitern oder kürzen.

Damit man auch bei großen Projekten den Überblick behält, ist eine Make-Utility integriert. MAKE aktualisiert nach einer Änderung in einer Unit alle betroffenen Projekte.

Alte Turbo-Pascal-Programme auf Turbo 4.0 umstellen

Programme, die keine computerspezifischen Anweisungen enthalten, laufen sofort auch unter Turbo 4. Bei den anderen Programmen sind Änderungen notwendig wobei die mitgelieferte Utility UPGRADE eine große Hilfe darstellt. Das Programm fügt die neuen Compileranweisungen ein (die alten Direktiven sollte man auf jeden Fall erst einmal löschen) und markiert alle kritischen Stellen durch Kommentarzeilen, die einen Änderungsvorschlag enthalten. Die meisten Probleme treten bei maschinennahen Programmteilen auf. So ist die Registervariable nun vordefiniert. Die Geräte CON:, AUX:, TRM: etc. gibt es nicht mehr, sondern nur noch die DOS-Geräteteien CON, AUX, PRN, etc. Dafür kann man jetzt Treiber für Geräte schreiben, die einer Textdatei zugeordnet werden (das sind aber keine MS-DOS-Einheitentreiber). Inline-Prozeduren müssen auf jeden Fall geändert werden, da die Übergabekonventionen von Parametern geändert wurden und vordefinierte Konstante nicht mehr im Code-Segment, sondern im Daten-Segment liegen. Außerdem akzeptiert der Inline-Code keine relativen Sprünge und keine Prozedur- oder Funktionsaufrufe mehr. Alternativ zum Inline-Code-Verfahren kann man jedoch OBJ-Dateien des Assemblers einbinden.

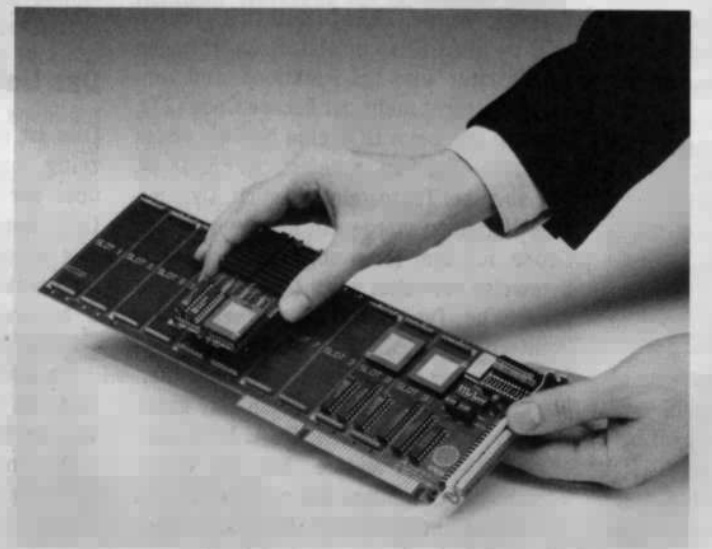
Umsteiger aufgepaßt

Eines läßt sich absehen – mit der Version 4 wird sich Turbo-Pascal noch weiter verbreiten als dies schon der Fall ist. Der neue Compiler dürfte vor allem jene ansprechen, die professionelle Programme entwickeln. Wenn man umsteigen will, kann man seine alte Version mit Handbuch an Heimsoeth einschicken und erhält die Version 4 für 170 Mark. Wer aber schon viele Programme mit der Version 3 geschrieben hat und die Umstellung scheut, sollte sich die neue Version zusätzlich zulegen. □

Transputer-Module

Unter dem Namen TRAM (Transputer Module) stellte Inmos in München eine Familie von Platinen vor, auf welchen Transputer-Chips und deren Peripherie in hoher Packungsdichte verschaltet sind. Grund- und Zusatzkarten mit unterschiedlichen Transputerkonfigurationen und unterschiedlichem Speicherausbau bilden ein System, das im Aufbau sehr flexibel ist. Die Grundkarte dient als Interface zwischen der Welt der Transputer und dem Host (es gibt Versionen für den IBM-PC, Apollo Workstation, Sun Workstation und VME-Bus). Auf diese Grundkarte, die selbst einen Transputer enthält, können mehrere TRAM-Bausteine aufgesteckt werden. Auf der Grundplatine ist ein sogenanntes „Softwire"-Netzwerk aufgebaut, das die Verbindungsstruktur des Transputer Arrays per Software einstellt. Ohne Umlöten oder Umstecken können damit Matrix-, Pipeline- oder Baumstrukturen als Verbindungsschema zwischen den Transputer-Modulen erzeugt werden. Dem Transputer ideal auf den Leib geschnitten, können dessen Vorteile mit den neuen Boards richtig zur Geltung gebracht werden. TRAM-Bausteine werden im Gebiet der Multi-Prozessor-Systeme und deren Anwendung und Konstruktion eine ähnliche Rolle spielen, wie es die klassischen Bus-Konzepte (VME oder Multibus) auf dem Gebiet der Mikrorechner heute schon tun, sagte Inmos. Das Bild zeigt eine Grundkarte für einen Rechner der AT-Klasse, auf der bis zu 10 Transputer-Baugruppen (schmale Bauform) Platz haben. Es gibt von den aufsteckbaren „Dual-Inline-Transputer-Baugruppen" auch breitere Varianten mit mächtigerem Speicherausbau. Ebenfalls im Frühjahr 1988 bietet Inmos neue Compiler für die Sprachen C, Fortran und Pascal an. Außerdem ein neues Produkt, das „IMS D705 OCCAM Toolset", das für die Entwickler wissenschaftlicher Sprachen konzipiert ist und Compiler Linker und Konfigurierer für den IBM-PC in einer TRAM-Hardware-Umgebung umfaßt. Damit können Quellen, Bibliotheken und Prozesse separat compiliert werden, die in einer der im Wissenschaftsbetrieb bereits eingeführten Sprache geschrieben wurden.

Man ist so in der Lage, parallele Strukturen auch in Programme einzubringen, die sonst von einem Lauf auf einem Transputersystem nur träumen könnten.



Durch Software einstellbar: Transputerstrukturen mit TRAMs aufbauen

Peter Wollschlaeger

Quick-C von Microsoft

Professionelles C für MS-DOS so einfach wie Basic

Fast unmerklich hat sich MS-C in den letzten Jahren als Industrie-Standard etabliert. Quick-C öffnet diese bisher geschlossene Welt der Profis den Einsteigern. Die Lösung ist eine Entwicklungsumgebung, wie man sie von Basic her kennt, genauer, wie in Quick-Basic.

Vorab: Den ersten Artikel über Quick-C habe ich für „etc", die Hauszeitschrift von Microsoft geschrieben, woraus Sie schließen können, daß ich als Beta-Tester fungiere, was richtig ist. Desweiteren könnten Sie schließen, daß ich nicht objektiv bin, was ich entschieden bestreite. Die Sache ist nämlich ganz einfach: Ich bekomme von zahlreichen Software-Häusern Beta-Versionen von neuen Programmen zur Beurteilung zugeschickt. Wenn sie was taugen, schreibe ich meistens Artikel darüber, wenn nicht, sage ich, entwickelt die Sache zu Ende und meldet Euch dann wieder.

Dem erfahrenen Anwender der Version 4.0 von MS-C würde es an dieser Stelle wahrscheinlich reichen, wenn man die Erweiterungen der Version 5.0 schildert und noch etwas auf die neue Bedienoberfläche eingeht. Es gibt jedoch viele Programmierer, die inzwischen eingesehen haben, daß man an C nicht mehr vorbeikommt. Diese Anwender suchen den bequemen Einstieg in diese Sprache. Tatsächlich sind die konventionellen Entwicklungspakete zwar extrem leistungsfähig, für den Einsteiger aber schon wegen der Fülle der Leistungsmerkmale verwirrend.

Diesen Markt sehr wohl erkennend, versucht Borland zur Zeit mit Turbo-C einen C-Volks-Compiler durchzusetzen, der dem ersten Anschein nach gar nicht schlecht ist. Doch mein erster Eindruck ist: Quick-C ist besser.

Quick-C ist nicht Microsofts Antwort auf Turbo-C, denn die Entwicklung wurde schon viel früher begonnen und die meisten H-Dateien tragen Copyright-Vermerke von 1984 bis 1987, sprich, sie stammen aus MS-C, Version 3, 4 und 5. Dennoch ist Quick-C nicht MS-C Version 5.0, sondern wird da als Zugabe geliefert. Man kann es aber auch einzeln, sogar recht preiswert, erwerben. Wieweit man dann noch MS-C braucht, ist eine andere Frage. Hier spielen dann Dinge wie die Entwicklung im Team (Version 5 ist netzwerkfähig) oder das Einmischen anderer Sprachen (Pascal, Fortran, Assembler) hinein.

Immerhin ist Quick-C auch als Tool für die Prototypen-Entwicklung gedacht, und muß deshalb schon alle Sprachelemente der großen Systeme beherrschen.

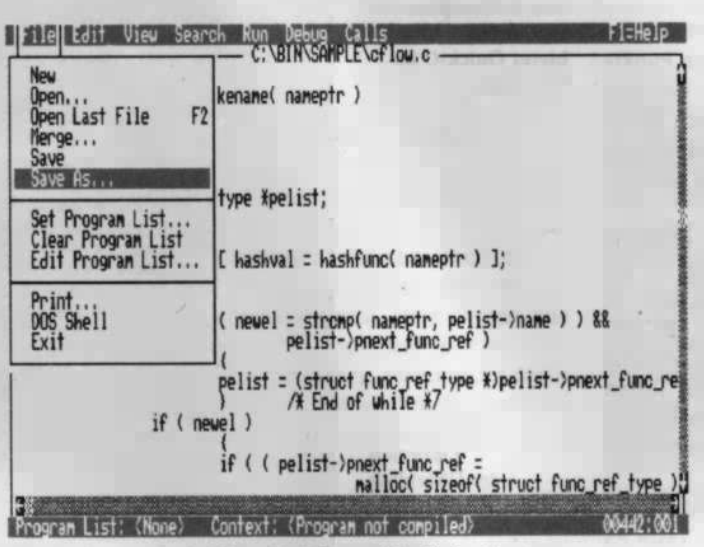


Bild 1. Quick-C wird über Pull-Down-Menüs bedient

Einfache Installation

Quick-C wird auf fünf nicht kopiergeschützten Disketten geliefert. Je nach Hardware-Konfiguration und nach persönlichen Vorstellungen muß der Anwender ein Programm namens SETUP mit mindestens drei Parametern aufrufen. Danach installiert sich das System selbst. Das Programm prüft, ob die angegebenen Parameter realistisch sind. So kann man zwar auf einem System mit Coprozessor bestimmen, daß man den 8×87 emuliert haben möchte, jedoch nicht Coprozessor-Unterstützung verlangen, wenn dieser Chip nicht bestückt ist.

Als Tester probiert man ja alles mögliche aus, um so ein Programm zu überlisten, aber sogar das Einlegen von Diskette 2, wenn Diskette 3 verlangt wird, bemerkt das Programm.

Unter anderem werden bei der Installation zwei Dateien generiert, die man in CONFIG.SYS bzw. AUTOEXEC.BAT einfügen sollte. Gespannt auf einen tollen Treiber sah ich mir also die CONFIG-Empfehlung an, fand hier aber nur „Files=20 Buffers=10". Immerhin beweist dies, daß man hier sehr sorgfältig darauf achtet, daß auch Anfänger nicht in Fallen laufen, was auch die Texte für die AUTOEXEC.BAT-Datei beweisen. Hier wird mit PATH und SET das „Environment" angepaßt.

Bedienung auch mit der Maus

So vorbereitet, reicht nun die Eingabe von „QC", und es erscheint der Editor *(Bild 1)*, natürlich noch ohne den Text und das heruntergezogene Menü. Letzteres wird man wohl immer als erstes wählen, denn einen Text muß man wohl laden oder eintippen, um weitermachen zu können. Man landet mit dem Aufruf von QC also immer im Editor.

Wer Quick-Basic kennt, wird sich gleich zu Hause fühlen, denn die Bedienoberflächen sind sehr ähnlich, die von QC hat nur mehr Funktionen und ist in einigen Punkten (noch) perfekter gestaltet. Ein Menü wird mit der Maus angeklickt. Hat man diese nicht, tippt man „Alt" plus den ersten Buchstaben des Titels. Innerhalb eines Menüs kann man nun dessen Items (Menüpunkte) mit den Cursor-Tasten anfahren oder den je Item hervorgehobenen Buchstaben tippen. Es reicht auch, wenn man ein beliebiges Menü anwählt. Von dort kann man mit den horizontalen Cursor-Tasten in die anderen Menüs wechseln.

Im Gegensatz zu Turbo-C haben die Menüs keine Untermenüs, und diese dann auch

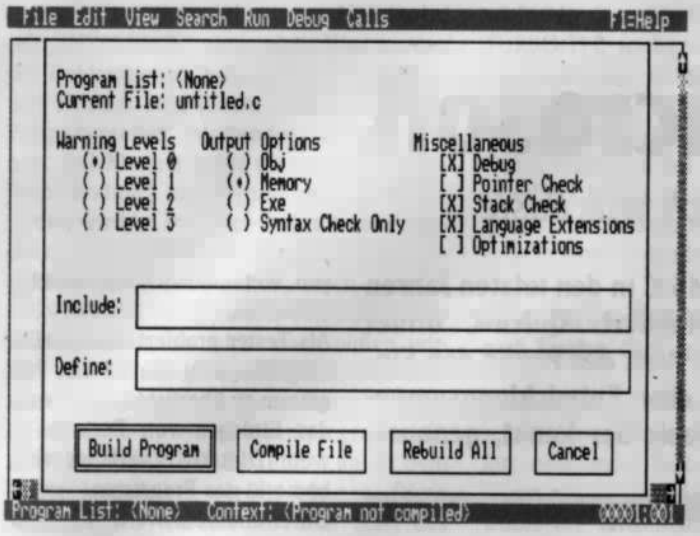


Bild 2. Compiler-Optionen bietet Quick-C im Überfluss

```
        QUICK C COMPILER OPTIONS
-MEMORY MODEL-
/AS small model (default)
/AC compact model
/AM medium model
/AL large model
-OPTIMIZATION-
/O enable optimization (same as /Ot)
/Od disable optimizations
/Ox enable optimization (same as /Ot)
/Ot enable optimization
-CODE GENERATION-
/G0 8086 instructions (default)
/G2  286 instructions
/Gc Pascal style function calls
/Gs no stack checking
/Gt[number] data size threshold
-OUTPUT FILES-
/Fe<executable file>
/Fl[object listing file]
/Fm[map file]
/Fo<object file>
-PREPROCESSOR-
/C don't strip comments
/D<name>[=text] define macro
/E preprocess to stdout
/EP same as /E but no #line
/I<name> add #include path
/P preprocess to file
/U<name> remove predefined macro
/u remove all predefined macros
/X ignore "standard places"
-LANGUAGE-
/Za disable extensions
/Zd line number information
/Ze enable extensions (default)
/Zi symbolic debugging information
/Zl remove default library info
/Zp pack structures
/Zq enable debug interrupts
/Zr enable pointer checking
/Zs syntax check only
-FLOATING POINT-
/FPi inline with emulator (default)
/FPi87 inline with 8087
-MISCELLANEOUS-
/c compile only, no link
/W<number> warning level
-LINKING-
/F<hex_number> stack size (hex. bytes)
/link [linker_options_and_libraries]
```

Bild 2a. Alle Optionen im Überblick

keine Unter-Unter-Menüs. Stattdessen werden Dialog-Boxen angeboten, deren Punkte man mit den Cursor-Tasten oder der Tab-Taste erreicht, zum Teil auch durch Eingabe des ersten Buchstabens des Feldnamens. Alle Funktionen sind mit der Maus oder der Tastatur erreichbar.

Man kann das Programm sehr gut ohne die Maus bedienen, wer sich allerdings an das Nagetier erst einmal gewöhnt hat, wird es auch nicht mehr missen wollen. Es gibt Fälle, in der die Maus eindeutig Vorteile hat, andere, in denen man mit den Tasten schneller ist. Am besten ist, man hat die Maus, für viele andere Programme braucht man sie ja sowieso.

Nun, ich habe erst einmal „printf("Hello World")" eingetippt und dann die Taste F5 betätigt, was den Compiler und Linker startet. Aus alter Gewohnheit (habe mal C mit einem Lattice-Compiler gelernt) lehnte ich mich nun zurück, um die wohlverdiente Pause einzulegen, aber das „war wohl nix". Kaum hatte ich nämlich wieder einen Blick auf den Bildschirm geworfen, stand da schon „Hello World". Da hast Du wohl geschlafen, dachte ich, also noch einmal. Diesmal paßte ich genau auf und siehe da: kaum hatte ich F5 betätigt, lief auch schon das Programm. Compiler- und Linker-Zeiten waren tatsächlich nicht festzustellen.

Sagenhaft schnell

So etwas gilt natürlich nur für solche Kurzprogramme. Bei längeren Texten kann man schon sehen, wie der Compiler rast, weil dann eine Anzeige (Zeilenzähler) mitläuft. Unter optimalen Bedingungen (AT und Festplatte) schafft der Compiler 10 000 Zeilen pro Minute. Borland gibt übrigens für sein Turbo-C 7 000 Zeilen/Minute an. Ich habe Quick-C allerdings unter ziemlich schlechten Bedingungen getestet. Mein Test-Computer war ein Amiga 2000 mit PC-Karte. In dieser Konfiguration läuft ein 8088 mit 4,77 MHz und allem Zubehör (RAM und so) als Hardware, das Display ist allerdings nur ein Amiga-Task (unter anderen) und nicht besonders schnell. Unter diesen Bedingungen bin ich handgestoppt auf 3 000 Zeilen pro Minute gekommen. Für 10 000 Zeilen pro Minute bräuchte man demnach eine Maschine mit einem Norton-Faktor von 3.3, was heutzutage nichts Besonderes mehr ist. Übrigens sagt ja so mancher, daß die Compiler-Zeiten nicht so relevant sind, weil man den Job mit SPAWN ja auch im Hintergrund laufen lassen und solange etwas anderes machen kann. Um ehrlich zu sein: Mir fällt für diese Pause meistens keine gescheite Alternativ-Tätigkeit ein.

Die Compiler-Optionen lassen sich laut *Bild 2* sehr bequem einstellen. Es gibt noch eine Auswahl dieser Art, in der man unter anderem eine Kommandozeile und die Stackgröße vorgeben kann. Auch hier kann man mit der Maus die Punkte der Dialog-Box anklikken. Ohne Maus hangelt man sich mit der Tab-Taste in die richtige Spalte und innerhalb dieser mit den Cursor-Tasten auf den gewünschten Punkt. Die Hardcopy macht übrigens nicht deutlich, daß einzelne Buchstaben aufgehellt sind, über die man die Felder direkt anwählen kann. In diese Dialog-Box gelangt man über das Menü „RUN". Alle Compiler-Optionen sind in *Bild 2a* aufgeführt.

Debugging auf Source-Ebene

Im Debug-Menü findet man Punkte, die man in einem Code-Debugger erwarten würde, doch das funktioniert auf Source-Ebene. Erklärenswert ist „Watch". Das ist nicht etwa eine Uhr, sondern eine Methode, um Variable während des Programmlaufs in einem Watch-Window überwachen zu können, auch auf Source-Ebene. Genauso kann man „tracen" und Breakpoints setzen. Vorher wird man sich aber wohl um die allgemeinen Denk- und Tippfehler kümmern müssen. Quick-C merkt sich während

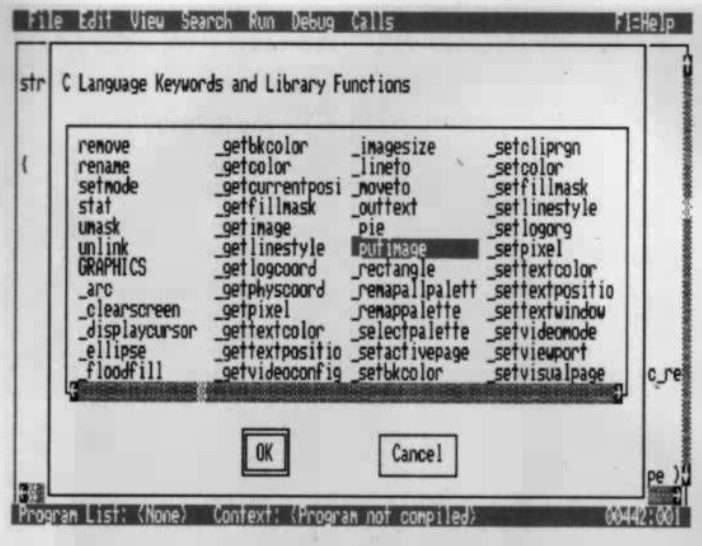


Bild 3. Quick-C ist sehr hilfreich. Aus dem Menü wird der zu erklärende Begriff ausgewählt

```
#include <stdlib.h>

#define ROWS 5000

fcmp(int *a, int *b)
{
  return(*a - *b);
}

main()
{
  int i, a[ROWS];

  for (i=0; i<ROWS; i++)
    a[i] = rand();

  printf("Start\n");

  qsort(a, ROWS, sizeof(int), fcmp);

  printf("Stop\n");

}
```

Bild 4. Eine Quick-Sort-Routine, die die eingebaute Funktion einsetzt, ist mit Quick-C schnell programmiert

des Compilerlaufs bis zu 26 Fehler. Danach kann man im Quelltext die Fehler der Reihe nach abrufen und sogar in dieser Folge wieder zurückschalten.

Wer mal schnell im Handbuch nachschauen möchte, braucht dazu kein Papier. Ein Druck auf die Taste F1 bringt Hilfe. Aus dem großen Angebot zeigt *Bild 3* eine Seite. Sehr nützlich sind die ASCII-Tabellen und die Liste mit dem umfangreichen Sprachschatz, der auch als kontext-orientierte Hilfe dient. Dazu stellt man im Editor den Cursor auf ein Schlüsselwort oder eine Lib-Funktion und drückt dann Shift-F1. Danach erscheint in einem Fenster der Aufruf mit der Syntax, den Datentypen, einer Kurzbeschreibung und dem Namen der H-Datei, die ggf. zu laden ist. Auch vordefinierte Strukturen kann man sich so auflisten lassen.

Läuft ein Programm, so arbeitet es mit einem eigenen Screen. Nach dem Programmende gelangt man mit einem Tastendruck wieder in den Editor zurück. Fällt einem danach ein, daß man wohl doch noch etwas von dem Programm-Screen sehen will, so ist das auch kein Problem. Mit F4 kann man zwischen den beiden Screens jederzeit umschalten. Man kann auch zwei Dateien auf zwei Screens halten, mit F2 bequem zwischen beiden umschalten und Textblöcke hin- und herkopieren.

C interaktiv wie Basic

Praktisch kann man mit Quick-C programmieren wie in Basic. Der Compiler bleibt zwar ein Compiler, aber dank seines hohen Tempos fallen die Zeiten zwischen den einzelnen „RUNs“ nicht mehr unangenehm auf. Dieses Feature dürften natürlich zuerst Einsteiger begrüßen, die durch die sonst ja sehr langwierige Strafschleife durch Editor, Compiler und Linker oft genug so frustiert werden, daß sie bald aufgeben und ihr Heil in immer moderneren Basic-Dialekten suchen. Sicherlich dürfte aber auch der C-Profi dieses Tool begrüßen, denn erstens machen Profis genauso viele Fehler wie Anfänger (sie brauchen dafür nur ein paar Programmzeilen mehr), zweitens ist damit eine sehr komfortable Prototyp-Entwicklung möglich, und drittens ist gerade für Profis Time auch Money.

Nun wollte ich natürlich wissen, wie schnell der Code ist. Microsoft selbst sagt, daß die Ablaufgeschwindigkeit ca. 30% über der von MS-C, Version 4.0 liegen wird. Das glaube ich erst einmal, und teste deshalb das Tempo im Vergleich zu Turbo-C. Das ist zwar etwas gewagt, denn immerhin liegt mir eine noch nicht freigegebene Beta-Version vor, aber probiert habe ich es doch. Die endgültige Release kann ja nur schneller sein. Begonnen habe ich mit der Ackermann-Funktion. Das Ergebnis hat mich etwas verblüfft. Quick-C erreichte nämlich in allen Varianten so präzise genau die selben Werte wie Turbo-C, daß man meinen könnte, die Routinen zur Stackverwaltung (die Funktion ist rekursiv) haben denselben Code. Übrigens muß der in einer anderen Zeitschrift zu MS-C angemerkte Stack-Overflow nicht sein. Man kann nämlich in C die Stackgröße einstellen, ab Akkerman(3,6) muß man sogar mehr als die voreingestellten 2048 Bytes opfern. Nun zum Sieve-Test, aber nur bis 5001. Hier wurde für Turbo-C 1.0 Sekunde gemessen.

```
#define ROWS 5000

sort(int liste[], int a, int e)
{
int i,j,s1;          /* Quicksort */
                     /* rekursiv  */
i=a; j=e;
s1=liste[a];
while (i<j)
  {
    while (i < j && s1 < liste[j])
     j--;
     if (i < j)
       {
         liste[i] = liste[j];
         i++;
       }
    while (i<j && liste[i]<=s1)
     i++;
    if (i < j)
      {
        liste[j] = liste[i];
        j--;
      }
  }
  liste[i]=s1;
  if (i-a )  sort(liste,a,i-1);
  if (e-j )  sort(liste,j+1,e);
}

main()
{
int i, a[ROWS];

for (i=0; i<ROWS; i++)
  a[i] = rand();

printf("Start\n");

sort(a,0,ROWS-1);

printf("Stop\n");

}
```

Bild 5. Diese QuickSort-Routine wurde selbst programmiert

Auch hier kam nur heraus, daß Quick-C auch diese Übung in derselben Zeit schaffte. Die Zeiten sind handgestoppt. Vielleicht mag eine genaue Messung ein paar Prozent Unterschiede ergeben, ich jedenfalls habe das Thema wegen „mangelnder Erfolgsaussicht" nicht weiter verfolgt. Jedenfalls kann man daraus nur folgern, daß ein Compiler nicht mehr kann, als den Quelltext in Maschinen-Code zu übersetzen. Diese Aufgabe kann man zwar mehr oder weniger gut lösen, inzwischen dürfte aber das Thema C-Compiler schon derart intensiv erforscht sein, daß es da nicht mehr viel zu holen gibt. Andererseits ist so ein Compiler natürlich ein sehr komplexes Gebilde. Hinzukommt, daß die zahlreichen Library-Funktionen, besonders die, die über den Standard hinausgehen, immer noch eine Quelle für Bugs und mehr oder weniger gute Lösungen sein können. Hier ist allerdings das meiste nicht neu, sondern Bewährtes aus den Vorgänger-Versionen.

Um nun trotzdem so eine Lib-Funktion zu testen, habe ich mir einmal den Quick-Sort vorgenommen. *Bild 4* zeigt, wie man diese Funktion (etwas unsauber, aber es geht) anwendet. Im ersten Ansatz erntete ich einen Runtime-Error mit der Meldung, daß der Stack überlief. Das ließ sich schnell korrigieren, es blieb aber die Erkenntnis, daß die Routine rekursiv arbeitet. Folglich habe ich laut *Bild 5* einen rekursiven Quick-Sort geschrieben, und zwar so, wie ich meine, daß er am schnellsten läuft. Tatsächlich sortierte meine Routine 5 000 zufällig erzeugte Zahlen in 6,5 Sekunden, die Lib-Funktion brauchte dafür 11 Sekunden. Kurz bevor ich mir nun auf die Schulter klopfte, fiel mir aber noch ein, daß ein Quick-Sort zum Bubble-Sort degeneriert, wenn man ihm eine bereits sortierte Folge anbietet. Damit getestet (Zahlen von 0-4999) brauchte meine Routine dann plötzlich 26 Sekunden und die Quick-C-Routine weniger als eine. Daraus folgt, daß die eingebaute Routine intelligenter ist. Das kostet natürlich Zeit, genauso wie ihre Flexibilität, die zur Folge hat, daß ständig die Vergleichsfunktion (hier fcmp) aufgerufen werden muß, wo ich direkt die Array-Elemente anspreche.

Bliebe noch die Frage, wie schnell man zu einem ausführbaren Programm kommt. Nun, es kommt darauf an, um mit Radio Eriwan zu antworten. Im Compilerlauf ist Quick-C schneller als Turbo-C, was aber nicht so sehr auffällt. Dann ist aber Quick-C schon fertig, und bei Turbo-C wuselt immer noch der Linker auf der Diskette herum. Diese Aussage gilt allerdings nur, wenn man im Standard-C bleibt. Geht man darüber hinaus, muß man bei Quick-C auch

```
C_KEYWORDS
break
continue
do
for
goto
if
return
switch
while
DIRECTIVES
#define
#undef
#include
#ifdef
#ifndef
#if
#line
PRAGMAS
alloc_text
check_pointer
check_stack
function
intrinsic
loop_opt
pack
same_seg
BUFFER_MANIPULATION
memccpy
memchr
memcmp
memcpy
memicmp
memmove
memset
movedata
CHARACTER_CLASSIFICATION_AND_CONVERSION
is...
isalnum
isalpha
isascii
iscntrl
isdigit
isgraph
islower
isprint
ispunct
isspace
isupper
isxdigit
toascii
tolower
_tolower
toupper
_toupper
DATA_CONVERSION
atof
atoi
atol
ecvt
fcvt
gcvt
itoa
ltoa
strtod
strtol
strtoul
ultoa
DIRECTORY_CONTROL
chdir
getcwd
mkdir
rmdir
FILE_HANDLING
access
chmod
chsize
filelength
_makepath
_splitpath
fstat
isatty
locking
mktemp
remove
rename
setmode
stat
umask
unlink
GRAPHICS
_arc
_clearscreen
_displaycursor
_ellipse
_floodfill
_getbkcolor
_getcolor
_getcurrentposition
_getfillmask
_getimage
_getlinestyle
_getlogcoord
_getphyscoord
_getpixel
_gettextcolor
_gettextposition
_getvideoconfig
_imagesize
_lineto
_moveto
_outtext
_pie
_putimage
_rectangle
_remapallpalette
_remappalette
_selectpalette
_setactivepage
_setbkcolor
_setcliprgn
_setcolor
_setfillmask
_setlinestyle
_setlogorg
_setpixel
_settextcolor
_settextposition
_settextwindow
_setvideomode
_setviewport
_setvisualpage
_wrapon
I/O_CONSOLE_AND_PORT
cgets
cprintf
cputs
cscanf
getch
getche
inp
inpw
kbhit
outp
outpw
putch
ungetch
I/O_LOW_LEVEL
close
creat
dup
dup2
eof
lseek
open
read
sopen
tell
write
I/O_STREAM
clearerr
fclose
fcloseall
fdopen
feof
ferror
fflush
fgetc
fgetchar
fgetpos
fgets
fileno
```

Bild 6. Komplette Liste der Quick-C-Funktionen

```
flushall
fopen
fprintf
fputc
fputchar
fputs
fread
freopen
fscanf
fseek
fsetpos
ftell
fwrite
getc
getchar
gets
getw
printf
putc
putchar
puts
putw
rewind
rmtmp
scanf
setbuf
setvbuf
sprintf
sscanf
tempnam
tmpfile
tmpnam
ungetc
vfprintf
vprintf
vsprintf
MATH
acos
asin
atan
atan2
bessel
cabs
ceil
_clear87
_control87
cos
cosh
dieeetomsbin
dmsbintoieee
exp
fabs
fieeetomsbin
floor
fmod
fmsbintoieee
_fpreset
frexp
hypot
j0
j1
jn
ldexp
log
log10
matherr
modf
pow
sin
sinh
sqrt
_status87
tan
tanh
y0
y1
yn
MEMORY_ALLOCATION
alloca
calloc
_expand
_ffree
_fheapchk
_fheapset
_fheapwalk
_fmalloc
_fmsize
free
_freect
halloc
_heapchk
_heapset
_heapwalk
hfree
malloc
_memavl
_memmax
_msize
_nfree
_nheapchk
_nheapset
_nheapwalk
_nmalloc
_nmsize
realloc
sbrk
stackavail
PROCESS_CONTROL
abort
atexit
execl
execle
execlp
execlpe
execv
execve
execvp
execvpe
exit
_exit
getpid
onexit
raise
signal
spawnl
spawnle
spawnlp
spawnlpe
spawnv
spawnve
spawnvp
spawnvpe
system
SEARCHING_AND_SORTING
bsearch
lfind
lsearch
qsort
STRING_MANIPULATION
strcat
strchr
strcmp
strcmpi
strcpy
strcspn
strdup
stricmp
strerror
_strerror
strlen
strlwr
strncat
strncmp
strnicmp
strncpy
strnset
strpbrk
strrchr
strrev
strset
strspn
strstr
strtok
strupr
SYSTEM_CALLS(DOS)
bdos
_disable
_dos_allocmem
_dos_close
_dos_creat
_dos_creatnew
_dos_findfirst
_dos_findnext
_dos_freemem
_dos_getdate
_dos_getdiskfree
_dos_getdrive
_dos_getfileattr
_dos_getftime
_dos_gettime
_dos_getvect
_dos_keep
_dos_open
_dos_read
_dos_setblock
_dos_setdate
_dos_setdrive
_dos_setfileattr
_dos_setftime
_dos_settime
_dos_setvect
_dos_write
dosexterr
_enable
_farjmp
_OFF
_SEG
_harderr
_hardresume
_hardretn
int86
int86x
intdos
SYSTEM_CALLS(BIOS)
_bios_serialcom
_bios_disk
_bios_equiplist
_bios_keybrd
_bios_memsize
_bios_printer
_bios_timeofday
TIME
asctime
clock
ctime
difftime
ftime
gmtime
localtime
mktime
_strdate
_strtime
time
tzset
utime
VARIABLE-LENGTH_ARGUMENT_LISTS
va_arg
va_end
va_start
va_alist
va_dcl
MISCELLANEOUS
abs
assert
div
getenv
ldiv
longjmp
_lrotl
_lrotr
perror
putenv
rand
_rotl
_rotr
_searchenv
setjmp
srand
swab
STRUCTURES
BYTEREGS
complex
diskfree_t
diskinfo_t
div_t
dosdate_t
DOSERROR
dostime_t
exception
FILE
find_t
_heapinfo
ldiv_t
rccoord
REGS
SREGS
stat
timeb
tm
utimbuf
videoconfig
WORDREGS
xycoord
```

den Linker bemühen, der dann auch herumsucht. Die Unterschiede werden um so krasser, je weniger Bibliotheksfunktionen benötigt werden.

Mißt man einmal die Zeit, die zwischen dem Compiler-Aufruf im Editor und dem Start des Programms liegt, so sind das bei einem einfachen Programm wie „Hello World" bei Turbo-C 8 Sekunden, bei Quick-C weniger als 1 Sekunde. Diese Zeiten gelten für eine Festplatte, bei Disketten wird der Unterschied noch größer.

Der Linker von Quick-C ist ein eigenständiges Programm. Da es aber einen DOS-Shell-Aufruf gibt, muß man Quick-C nicht aufgeben, wenn man den Linker braucht. Gleiches gilt für die Make-Utility.

Meiner Meinung nach arbeitet man bei größeren Projekten am besten mit Make und editiert und testet die einzelnen Module mit Quick-C.

Zum Schluß zu *Bild 6*. Diese wahrlich lange Liste zeigt, was Quick-C bietet. Wenn da nicht auch die vielen DOS- und BIOS-Funktionen wären, könnte man glatt meinen, man ist gar nicht auf einem DOS-PC, sondern auf einem Unix-System. Wie *Bild 5* schon zeigt, beherrscht Quick-C auch die ANSI-Erweiterungen. Wer will, kann natürlich konventionell programmieren, und damit auf die strenge Typprüfung verzichten. Alles in allem schildere ich hier nur erste Eindrücke, doch die sind beeindruckend. Für eine Beta-Version ist das Paket schon erstaunlich perfekt. Jedenfalls hat es allerhand trickreiche Programme anstandlos geschluckt. Wenn die erste offizielle Release und die endgültige Dokumentation vorliegt, werden wir darüber berichten. □

Dieter Strauß

Grafik mit der VEGA VGA

Farbgrafiken waren auf PCs lange Zeit unbefriedigend darstellbar. Erst seit der Einführung der EGA kann man diese auf dem PC in akzeptierbarer Qualität erstellen. Eine weitere Verbesserung bringen Karten nach dem VGA-Standard. mc hat die VEGA VGA von Video Seven ausführlich getestet.

Bei der EGA ist die Zahl der gleichzeitig darstellbaren Farben stark begrenzt. Dies liegt nicht nur an der Konstruktion der EGA, sondern auch an den verwendeten Monitoren mit TTL-Eingängen. Bei ihnen ist die Zahl der gleichzeitig darstellbaren Farben auf die Zweierpotenz der Anzahl der Farbsignalleitungen beschränkt. Wegen dieser Limitierung hat IBM die PS/2-Serie mit Analog-Monitoren ausgestattet und zur Erhöhung der Auflösung den VGA-Standard eingeführt, der maximal 640 × 480 Bildpunkte auf dem Bildschirm darstellt.
Einer der ersten Anbieter von VGA-kompatiblen Karten ist das kalifornische Unternehmen Video Seven, das sein Produkt VEGA VGA (*Bild 1*) nennt. In Deutschland werden alle Grafikkarten von Video Seven über Computer 2000, München, vertrieben.

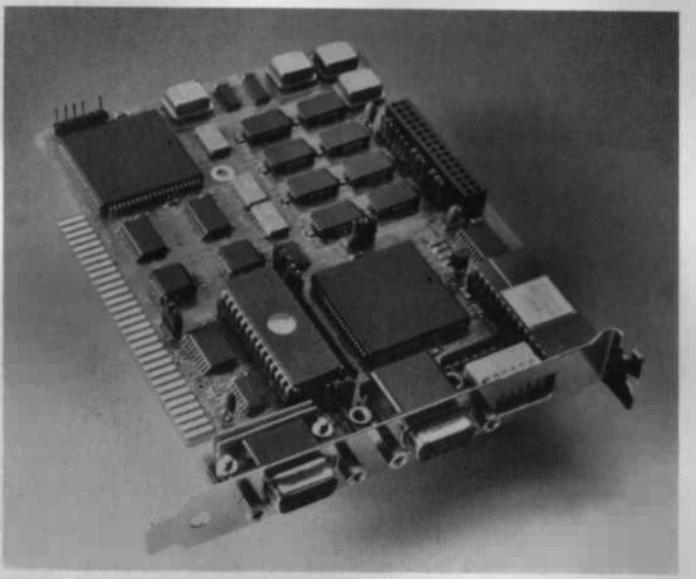


Bild 1. Die VEGA VGA ist wegen der zahlreichen Bauteile in platzsparender SMD-Technik als kurze Einsteckkarte ausgeführt

Die VEGA VGA ersetzt folgende Adapter:
- MDA (Monochrome Display Adapter)
- HGC (Hercules Graphics Card)
- CGA (Color Graphics Adapter)
- EGA (Enhanced Graphics Adapter)
- VGA (Video Graphics Adapter)

Wie man sieht, ist in der „Ahnenreihe" alles vertreten, was Rang und Namen hat. Zusätzlich – sozusagen als Bonbon – kennt die Karte ein paar weitere Betriebsarten.
Bild 2 gibt einen Überblick der verschiedenen Betriebsarten. Mit einer Auflösung von bis zu 800 × 600 Bildpunkten übertrifft die VEGA VGA den VGA-Standard bei weitem. Im Gegensatz z. B. zur Sigma-VGA ist sie fast durchweg mit platzsparenden SMD-Bausteinen (Surface Mounted Devices: Bauteile in Oberflächenmontagetechnik) bestückt. Bevor wir auf die Testergebnisse eingehen, sei zunächst die Technik der Karte geschildert.
Das Herz der Karte ist der VGA-Chipsatz, der aus zwei Bausteinen, dem Grafik-/Attribute-Chip und dem Sequenzer-/CRT-Controller, besteht. Der Rest der Schaltung (*Bild 3*) besteht nur noch aus den Speichern für das Video-RAM, dem BIOS, der Busschnittstelle, den Quarzoszillatoren und als wesentliches Teil der Farbpalette IMS-G171, die von der Transputer-Firma INMOS gefertigt wird. Mit der Farbpalette ist es möglich, 256 Farben aus einer Palette von 262 144 Farben gleichzeitig darzustellen. Wie aus Bild 2 ersichtlich ist, sind die 256 Farben nur im niedrigauflösenden Modus (320 × 200) darstellbar, da der Speicherplatz auf der VEGA VGA wie bei der IBM-VGA auf 256 KByte begrenzt ist. Bei 256 darstellbaren Farben muß schließlich für jeden Bildpunkt ein Byte reserviert werden.
Trotz der relativ niedrigen Auflösung sehen Bilder im 256-Farben-Modus gut aus und haben mit CGA-Bildern, trotz gleicher Auflösung, nichts mehr zu tun. Dazu dürfte auch die Double-Scan-Technik beitragen, die eine Bildzeile zweimal hintereinander auf dem Bildschirm darstellt. Bei unserem Test waren wir besonders von der Qualität von Schwarzweiß-Bildern überrascht, die schon einer Schwarzweiß-Fotografie mit grobem Korn nahekommen. Allerdings ist dazu unbedingt ein Analog-Monitor (z. B. Multisync) notwendig, denn nur er kann 256 Farben bzw. 256 Graustufen darstellen.
Wesentlich für die Geschwindigkeit des Bildaufbaus ist bei einer Grafikkarte die Breite des Video-Daten-Bus. Hier hat Video Seven nicht gespart und einen 32-Bit-Bus zwischen dem VGA-Chipsatz und dem Bildwiederholspeicher verwendet. Allerdings besteht nach wie vor das Nadelöhr der 8 Bit breiten PC-Bus-Schnittstelle, denn der Urahn aller PC-Video-Controller, der 6845, geistert auch in diesem Chipsatz herum. Der Zugriff auf dessen Ports ist prinzipiell nur in 8-Bit-Breite möglich. Lediglich der Datenpfad über den VGA-Chipsatz zum Video-RAM könnte bei einem kompatiblen Design in 16-Bit-Breite ausgeführt werden. Allerdings setzt eine solche Karte einen Rechner der AT-Klasse voraus. Als Besonderheit verfügt die VEGA VGA über eine Schaltung, die das Setzen und Löschen des Maus-Zeigers bei jeder Mausbewegung berechnet. Die System-CPU wird damit entlastet, jedoch muß diese Eigenschaft vom jeweiligen Maustreiber unterstützt werden.

Wichtig für Multitasking-Betriebssysteme sind die lesbaren Video-Register. Registerstände können damit im Speicher abgelegt und bei Bedarf wieder geladen werden, so daß für den Programmierer das Mitführen von Schattenregistern im RAM entfällt.

Die VEGA VGA kann in die Betriebsarten VGA/EGA, CGA oder MDA eingestellt werden oder sie kann automatisch in die jeweilig gewünschte Betriebsart gelangen. Der sogenannte Smartswitch erkennt die von der Anwender-Software angesprochenen E/A-Adressen, und schaltet auf die richtige Betriebsart um. Ebenso stellt die Karte den Typ des angeschlossenen Monitors fest. Hier muß man natürlich aufpassen, denn im mitgelieferten Handbuch steht ganz deutlich, daß sie den angeschlossenen Mo-

nitor nur richtig identifizieren kann, wenn dieser mit dem jeweiligen Original-IBM-Monitor im Verhalten übereinstimmt.
Im schlimmsten Fall könnte sogar der angeschlossene Monitor zerstört werden. Diese Gefahr ist jedoch keine Eigenheit der VEGA VGA sondern aller Karten, die den Monitortyp automatisch erkennen. Bei Monitoren, die sich automatisch auf die Zeilenfrequenz einstellen (z. B. NEC Multisync und Sony Multiscan) besteht natürlich keine Zerstörungsgefahr.

Während unseres Tests auf Kompatibilität mit der Hercules-Karte, den wir auf dem Commodore PC 10, dem Wyse-AT und dem mc-modular-AT durchführten, haben wir daher per Brücke und DIP-Schalter das System auf den Monochrom-Monitor eingestellt und auch tatsächlich einen angeschlossen. Sowohl bei Standard-Software (Wordstar, Euroscript, Harvard Graphics) als auch bei dem im PC-Sonderheft veröffentlichten Programm „Grafik mit der Hercules-Karte" [1] ergaben sich keine Probleme.

Der bei solchen Tests obligatorisch verwendete Microsoft Flugsimulator zeigte, daß auch der CGA-Modus funktioniert. Um den EGA-Modus zu testen, haben wir die Karte neben mehreren EGA-Demo-Programmen mit den Programmen aus unserer Serie „Grafik mit der IBM-EGA" [2] traktiert. Auch bei diesen Programmen, die zur Steigerung der Zeichengeschwindigkeit teilweise direkt auf Register zugreifen, versagte die Karte nicht. Den VGA-Modus haben wir mit dem Programm Harvard Graphics, das auf IBM VGA eingestellt war, überprüft. Interessant ist auch der Vergleich der Auflösung unter Autocad im Standard-VGA-Modus und im Modus mit einer Auflösung von 800 × 600 Bildpunkten. Im hochauflösenden Modus sind die „Treppen" bei schräg verlaufenden Linien fast kaum noch zu erkennen.

DIP-Schalter von außen bedienbar

Ein großes Lob verdienen die Konstrukteure, denn sie haben den DIP-Schalter zur Voreinstellung der Betriebsart nicht unzugänglich auf der Platine untergebracht, sondern von der Geräterückseite bedienbar gemacht. Man legt mit dem Schalter z. B. fest, ob die Karte im EGA/VGA- oder im CGA/MDA-Modus booten soll und kann auch die automatische Umschaltung auf den entsprechenden Bildschirm-Modus unterbinden. Sowohl ein Analog- als auch ein TTL-Ausgang ist verfügbar, an die man gleichzeitig je einen Monitor anschließen darf.

Bildschirm-Modus	Typ	Alpha-Format	Grafik-auflösung	Farben	Farbe/ Monochrom	Monitor
00 H	T	40 x 25		2	F, SW	C, E, V, 85xx
01 H	T	40 x 25		16	F	C, E, V, 85xx
02 H	T	80 x 25		16	F, SW	C, E, V, 85xx
03 H	T	80 x 25		16	F	C, E, V, 85xx
04 H	G		320 x 200	4	F	C, E, V, 85xx
05 H	G		320 x 200	2	F, SW	C, E, V, 85xx
06 H	G		640 x 200	2	F, SW	C, E, V, 85xx
07 H	T	80 x 25		4	M	M, V, 85xx
0D H	G		320 x 200	16	F	E, V, 85xx
0E H	G		640 x 200	4	F	E, V, 85xx
0F H	G		640 x 350	4	M	M, E, V, 85xx
10 H	G		640 x 350	16	F	E, V, 85xx
11 H	G		640 x 480	2	F, SW	V, 85xx
12 H	G		640 x 480	16	F	V, 85xx
13 H *	G		320 x 200	256	F	V, 85xx
40 H	T	80 x 43		16	F	E, V, 85xx
41 H	T	132 x 25		16	F	E *, V
42 H	T	132 x 43		16	F	E *, V
43 H	T	80 x 60		16	F	V, 85xx
44 H *	T	100 x 60		16	F	V, 85xx
4D H	T	120 x 25		16	F	E, V
4E H	T	120 x 43		16	F	E, V
4F H	T	132 x 25		16	F	C, E, V
50 H	T	80 x 43		4	M	M
51 H	T	132 x 25		4	M	M
52 H	T	132 x 43		4	M	M
60 H	G		752 x 410	16	F	V
61 H *	G		720 x 540	16	F	V
62 H *	G		800 x 600	16	F	V

* Diese Modi gibt es nur bei der VEGA VGA

Verwendete Abkürzungen:

F:	Farbe	C:	CGA-Monitor
M:	Monochrom (Graustufen darstellbar)	E:	EGA-Monitor
SW:	Schwarzweiß	V:	Monitor mit variabler Zeilenfrequenz
85xx:	IBM-Monitor Serie 85xx		

Bild 2. Zu den Standard-VGA-Betriebsarten hat Video Seven noch einige hinzugefügt

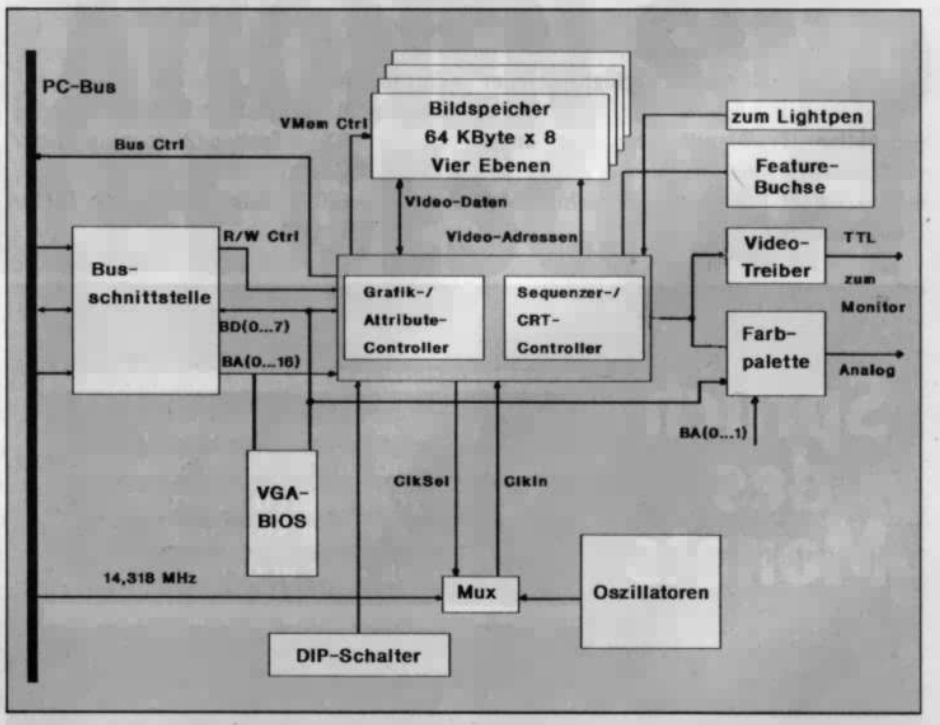


Bild 3. Blockschaltbild der VEGA VGA

Inklusive Software

Die VEGA VGA wird mit Hilfsprogrammen und Treibern für Autocad, Lotus, Symphony, Word Perfect, Wordstar, GEM und Windows geliefert. Die Treiber sind für die von Video Seven geschaffenen erweiterten Betriebsarten gedacht. Mit dem Hilfsprogramm VEGA.COM kann man z. B. einige Voreinstellungen des DIP-Schalter-Blocks unwirksam machen und einen Screen Saver installieren. Ein Screen Saver schützt den Bildschirm, indem er nach einer bestimmten Zeit ohne Tastenanschläge den Bildschirm dunkel schaltet. Im Handbuch weist jedoch Video Seven darauf hin, daß bei Programmen, die direkt die Tastatur abfragen, der Screen Saver den Bildschirm dunkel schaltet, obwohl der Anwender diese betätigt. Während unseres Testes ist dies genau bei Euroscript geschehen. In solchen Fällen darf man den Screen Saver nicht installieren oder man muß ihn vor dem Aufruf des betroffenen Anwender-Programms mit dem Kommando „VEGA SAVE: OFF" deaktivieren.

Programmierung der VEGA VGA

Video Seven hat das Standard-EGA/VGA-BIOS erweitert und einige Video-Betriebsarten hinzugefügt. Die Erweiterungen sind so implementiert, daß die Kompatibilität gegenüber der Original-EGA bzw. VGA erhalten bleibt. Solange IBM oder ein anderer Hersteller nicht andere Funktionen nach dem gleichen Aufrufschema einfügt, ist diese Methode akzeptabel.

Die vier Funktionen lauten:
- Funktion 0: Abfrage auf Verfügbarkeit der BIOS-Erweiterungen
- Funktion 1: Identifizieren des angeschlossenen Monitors
- Funktion 4: Erweiterten Bildschirm-Modus feststellen

AL	Monitor
0	nicht spezifiziert
1	IBM Monochrome Display
2	IBM Color Display
3	IBM Enhanced Color Display
4	NEC Multisync oder kompatibler Monitor
5	Sony Multiscan oder kompatibler Monitor
6	Magnavox Multimode
10	IBM PS/2 Analog oder kompatibler Monitor

Bild 4. Einige Monitore werden von der VGA automatisch erkannt

- Funktion 5: Erweiterten Bildschirm-Modus setzen

Vor dem Aufruf der erweiterten BIOS-Funktionen mit dem Interrupt 10H muß im AH-Register 6FH, in AL die Funktionsnummer und in BX eventuell weitere benötigte Werte stehen.
Bevor eine der anderen erweiterten BIOS-Funktionen aufgerufen wird, sollte immer zuerst geprüft werden, ob die VEGA VGA im System ist.
Die Funktion 0 (BIOS-Erweiterung verfügbar?) hat folgende Parameter:

Eingabe: AH = 6FH, AL = 0
Ausgabe: BH = 'V', BL = '7'

Wenn in BX nach dem Funktionsaufruf ein anderer Wert als 'V7' steht, ist keine VEGA VGA im System. Die erweiterten BIOS-Modi können damit nicht benutzt werden.
Die Parameter der Funktion 1 lauten:

Eingabe: AH = 6FH, AL = 1
Ausgabe: AL = Monitor-Typ (*Bild 4*)

Folgende Parameter werden von der Funktion 4 (erweiterten BIOS-Modus feststellen) benötigt:

Eingabe: AH = 6H, AL = 4H
Ausgabe: AL = aktueller Bildschirm-Modus
BX = Spalten/Bildpunkte (Text/Grafik)
CX = Zeilen/Bildpunkte (Text/Grafik)

Diese Funktion entspricht der Standard-BIOS-Funktion (AH = 0FH, INT 10H), wenn die Karte in einer Standard-Betriebsart läuft. Wird eine der erweiterten BIOS-Funktionen verwendet, gibt die Standard-BIOS-Funktion je nach Betriebsart die Werte 0 (Farbe/Text), 7 (erweiterter monochromer Modus) oder 13H (erweiterte Grafik) zurück.
Die Funktion 5 (Erweiterten Bildschirm-Modus setzen) kennt folgende Parameter:

Eingabe: AH = 6FH, AL = 5H, BL = Wert des Bildschirm-Modus

Außer den erweiterten BIOS-Funktionen kann man dieser Funktion alle Standard-Bildschirm-Modi von 0H...12H übergeben.

Zukunft der PC-Farbgrafik

Der EGA-Standard wird in Anwendungen, bei denen es auf qualitativ hochwertige Farbgrafiken ankommt, relativ schnell durch den VGA-Standard abgelöst. Die höherauflösenden Modi der VEGA VGA werden bereits von einigen Anwender-Programmen unterstützt, inwieweit er sich durchsetzt bleibt abzuwarten. Durch die weitgehende Kompatibilität mit EGA, CGA und HGC sind auch ältere Grafik-Programme ablauffähig.
Ein weiterer Pluspunkt ist das mitgelieferte Handbuch. Auf insgesamt 52 Seiten wird der Einbau der Karte und die Bedienung der Hilfsprogramme beschrieben. Außerdem wird für Programmierer geschildert – allerdings etwas knapp – wie man die erweiterten Video-Funktionen programmiert. Bis zum Zeitpunkt der Veröffentlichung dieses Testberichts soll das Handbuch auch in Deutsch vorliegen.
Nur wer noch größere Anforderungen an die Geschwindigkeit und Auflösung stellt, wird sich eine wesentlich teure Grafik-Karte mit einem speziellen Grafik-Prozessor zulegen müssen.

Literatur

[1] Mußmann, Volker: Grafik mit der Hercules-Karte. Das PC-Sonderheft, Seite 62. Franzis-Verlag, München.
[2] Cebulla, Ulrich: Grafik mit der IBM-EGA. mc 1987, Ausgabe 1...4.

Spruch des Monats

Es ist wie beim Kochen,
Spaghetti sind nicht gleich Spaghetti.
Aufs Programmieren übertragen:
wenn schon Spaghetti,
dann aber al dente.

Hans Knichel, Informatik-Lehrer, 6000 Saarbrücken

GANZE INFORMATIONEN SIND DAS HALBE GESCHÄFT

Wirtschaftlicher Erfolg hängt immer mehr davon ab, daß Sie aus der Flut von Informationen die für Ihr Unternehmen wichtigen herausfiltern, bewerten und für Ihre Entscheidungen nutzen.

Nur auf der Basis abgesicherter Informationen lassen sich heute die Entscheidungen treffen, durch die das Unternehmen morgen die nötigen Gewinne machen soll.

Im Rahmen der finanziellen Betreuung sehen wir daher auch in der Rationalisierung der Zahlungsvorgänge eine wichtige Aufgabe für die Zusammenarbeit mit unseren Kunden.

Das entlastet Sie von zeitraubenden Verwaltungsarbeiten.

Sprechen Sie also mit uns, wenn es um die Verbesserung von Gelddispositionen, um belegloser Datenträgeraustausch, Scheck- und Wechseleinzug, Lohn- und Gehaltsabrechnungen geht.

Damit Ihnen die Zeit bleibt, Unternehmer zu sein.

Michael Sperber

C-Compiler im Vergleich

Drei C-Compiler für den Atari ST

In jedem Jahr veranstaltet die Stiftung Warentest den Wettbewerb „Jugend testet". Unser Autor gewann 1987 den zweiten Preis mit dem Vergleichstest von drei C-Compilern für den Atari ST, den Sie hier lesen können. Die C-Compiler werden gründlich und objektiv beurteilt.

Mit Rechnern wie dem Sinclair QL und dem Apple Macintosh begann die Ära der 68000er-Rechner; es folgten der Atari ST und der Commodore Amiga. Die Rechner der neuen Generation übertrafen die Leistungen der „alten" 8-Bit-Computer bei weitem. Als Mikroprozessor wird der 68000 verwendet, der bis zu 16 MByte adressiert, einen leistungsfähigen Befehlssatz besitzt und mit 16 MHz getaktet werden kann.
Waren früher 64 KByte noch viel Speicher, so ist heute unter einem Megabyte kaum noch einer dieser Rechner zu haben. Diese Leistungsdaten galten noch vor wenigen Jahren als unerreichbar.
Auch die Anforderungen an die Software und deren Leistungsfähigkeit stiegen. Die Entwicklung leistungsfähiger Software erfordert allerdings mehr Arbeit, mehr Zeitaufwand und produziert damit mehr Kosten für den Entwickler. Auf den alten Computern waren professionelle Programme ausschließlich in Assembler erstellt worden, der „Muttersprache" des Mikroprozessors. Assemblerprogramme sind schnell und kompakt, was sie bei den damaligen Beschränkungen bezüglich Speicherplatz und Geschwindigkeit auch sein mußten. Doch in Assembler zu programmieren ist mühsam, weil jeder Befehl für sich sehr wenig bewirkt. Außerdem werden Assemblerprogramme schnell unübersichtlich.
Es lag nahe, komplexe Programme in Hochsprachen zu entwickeln, die der menschlichen Denkweise besser angepaßt sind. Sie kommen zwar in Effizienz und Kompaktheit Assemblern nicht ganz nach, erlauben jedoch eine schnellere und komfortablere Entwicklung. Unter den Hochsprachen hat sich C etabliert, weil in C sehr schnelle und kompakte Programme geschrieben werden können.
Die Sprache C erfreut sich großer Popularität, auch bei Heimanwendern und Hobby-Programmierern, besonders weil sie sich auf 68000-Rechner effizient umsetzen läßt. Bei den verschiedenen C-Compilern gibt es zum Teil gravierende Unterschiede. Es gibt schnelle und langsame, komfortable und spartanische, umfangreiche und kleine, vollständige und unvollständige, teuere und preiswerte Realisationen von C. Keineswegs ist die teuerste immer die beste.
Hier sollen drei C-Compiler untersucht werden, die wohl populärsten auf dem Atari ST. Dieser Testbericht entstand aus einer langen intensiven Arbeit mit allen getesteten Compilern, wobei auch größere Projekte realisiert wurden.

In den Kampfring, bitte

Getestet wurden der Alcyon-C-Compiler von Digital Research (DRI), der im von Atari vertriebenen Entwicklungssystem enthalten ist. Das Entwicklungssystem ist mit Dokumentation für etwa 1600 DM zu haben. Der zweite Kandidat ist der Lattice-C-Compiler, der von der britischen Firma Metacomco vertrieben wird. Er kostet zur Zeit etwa 300 DM. Auch der C-Compiler der amerikanischen Firma Megamax, der momentan ca. 450 DM kostet, mußte zeigen, was er kann.
Der DRI-C-Compiler hat keine Versionsnummer, offenbar gibt es nur eine einzige Version. Der Compiler von Lattice wurde in der Version 3.03, der von Megamax in der Version 1.0 getestet. Auf die Neuerungen in der Version 3.04 des Lattice-C-Compilers wird hingewiesen.

Lieferumfang

Das Entwicklungssystem von DRI ist am umfangreichsten. Dazu gehören 7 Disketten und mehrere Ordner mit zusammen ca. 2000 Handbuchseiten. In diesem System ist praktisch alles enthalten, was der Programmierer braucht: 3-Pass-C-Compiler, Editor, Assembler, Linker, Batch-Prozessor, Kommandoprozessor, GEM-Bibliotheken, Librarian, Debugger, Resource Construction Set, ein Malprogramm, die Quelltexte für die Runtime-Library und viele weitere Tools.
Beim Lattice-C-Compiler in der Version 3.03 sieht es da schon magerer aus: Zwei Disketten und ein etwa 250seitiges Handbuch. Im Lieferumfang ist der 2-Pass-Compiler, der Linker, ein Editor sowie eine GEM-Bibliothek mit Quelltext enthalten. Die Version 3.04 des Lattice-Compilers gibt es mit einem 600 Seiten starken deutschen Handbuch, Debugger, Resource Construction Set, GEM-Shell und zahlreichen neuen Routinen, die auch den Floating-point-Coprozessor 68881 unterstützen.
Der Megamax-Compiler wird auf zwei Disketten geliefert, zusammen mit einer ca. 350 Seiten starken Dokumentation. Auf der Diskette tummeln sich der 1-Pass-Compiler, der Linker, ein Editor, ein Librarian, ein Disassembler, ein „Code Improver", ein Resource Construction Programm, eine GEM-Bibliothek und eine „Graphic Shell".

Platzbedarf

Für den Atari ST-Anwender, der keine Festplatte hat, ist es von Wichtigkeit, auch ohne Festplatte Programme erstellen zu können. Er wird sich eine RAM-Disk mit 512 KByte anlegen, in der er Compiler und Programm ablegt. Beim Compiler von DRI gibt es da schon Schwierigkeiten, da er 350 KByte belegt. Da er außerdem noch sehr viel Platz für Zwischendateien benötigt, bleibt kaum noch Platz für das C-Programm. Häufig ist ein Auslagern von Dateien auf Diskette unvermeidlich.
Lattice-C paßt noch locker in 250 KByte der RAM-Disk. Auch 2000-Zeilen-Programme vertragen sich noch gut mit ihm, da er längst nicht soviele Zwischendateien auslagert wie der DRI-Compiler.
Noch genügsamer ist der Megamax-Compiler, der noch weniger Platz belegt und noch weniger Zwischendateien einrichtet.

Entwicklungsumgebung

Für den Programmierer ist es wichtig, den Entwicklungsvorgang steuern zu können. Zwar ist das Prinzip immer das gleiche (Editieren, Compilieren, Linken, Testen,

Editieren...), doch die Entwicklungsschritte können auf unterschiedliche Art und Weise unter einen Hut gebracht werden.
Der DRI-Compiler stellt einen Batch-Prozessor zur Verfügung. Mit ihm werden Compilierung und Linkvorgang zusammengefaßt, ohne daß der Benutzer separate Kommandos geben muß. Die dazugehörigen Programmaufrufe werden in eine Datei geschrieben, von wo der Batch-Prozessor sie dann ausführt. Außerdem wird ein Kommandoprozessor mitgeliefert, der über eine Kommandosprache die Steuerung des Systems übernimmt. Ein Batch-Prozessor ist hierin enthalten. Nicht jeder MS-DOS-Kenner und Unix-Freak wird sich sofort „wie zu Hause" fühlen, er kann sich aber doch recht schnell einarbeiten.
Beim Lattice-Compiler müssen in der Version 3.03 alle Programme vom GEM-Desktop aus einzeln gestartet werden, sowohl der Compiler, als auch Linker und Editor, was umständlich ist. Immerhin werden die beiden Compiler-Läufe von einem kleinen Steuerprogramm zusammengefaßt. In der neuen Version 3.04 wird die GEM-Shell MENU+ mitgeliefert, mit der die Einzelprogramme per Maus angesteuert werden, was die Arbeit sehr vereinfacht.
Der Megamax-Compiler hat einen besonderen Leckerbissen im Lieferumfang: Die sogenannte Graphic-Shell. Dieses (ebenfalls mausgesteuerte) Menüprogramm ermöglicht den Aufruf von allen Teilen des Megamax-Pakets auf einfache Art und Weise. Die Graphic-Shell ist dabei flexibler als das MENU+ des Lattice. Ein besonderer Vorteil ist, daß die Shell bei Fehlern im Programm die Fehlermeldungen in eine Datei schreibt und diese automatisch mit dem Editor lädt. Auf diese Weise wird die Fehlerkorrektur stark vereinfacht. Der Clou der Graphic-Shell besteht in der Option „Make", die für größere Programmprojekte sehr praktisch ist. Wenn ein großes Programm aus mehreren Modulen besteht, sorgt Make automatisch für die Compilierung und das Linken aller Module, die verändert wurden, ohne daß der Benutzer auch nur einen Finger rühren muß. Auch die Version 3.04 des Lattice-C wird mit einer Make-Utility geliefert.
Trotzdem: In dieser Disziplin bietet keiner der Compiler das Nonplusultra. Für den IBM-PC gibt es C-Compiler, die z. B. ein weit leistungsfähigeres „Make" bieten. Wer Erfahrungen mit Unix hat, wird dieses Feature besonders vermissen.

Der Editor

Auch C-Programme wollen geschrieben werden. Dazu ist der Editor da, der zur Eingabe und Korrektur dient. Editoren gibt es wie Sand am Meer. Für den Entwickler gilt natürlich: je komfortabler und schneller, desto besser.
Bei älteren Auslieferungen des Entwicklungspakets war der Editor MINCE (was übersetzt die sinnige Bezeichnung „Hackfleisch" ergibt) enthalten. Dieser Editor ist sehr leistungsfähig. Er besitzt viele Sprunganweisungen, speichert gelöschten Text zwischen, um ihn später wieder zurückholen zu können, kann mehrere Texte gleichzeitig verwalten und bietet vieles mehr. Die Tastenkommandos sind mit denen des EMACS identisch, einem auf Großrechnern sehr beliebten Editor. Allerdings erlaubt MINCE keine Mausbedienung, was Freaks kaum stören wird. Er hat nur zwei Nachteile: Er ist behäbig, und es kann schon mal einen Moment dauern, bis sich auf einen Tastendruck hin etwas rührt. Außerdem, was schwerer wiegt, verabschiedet sich der MINCE auf die Eingabe eines Umlautes oder eines „ß" auf Nimmerwiedersehen.
In jetzigen Auslieferungen des Entwicklungssystems und beim Lattice-Compiler wird der Metacomco-Editor ED mitgeliefert. Auch er ist auf Tastaturbedienung ausgelegt, jedoch nicht so leistungsfähig. ED unterstützt weder die gleichzeitige Bearbeitung mehrerer Texte, noch speichert er gelöschten Text zwischen. Dafür kann er mit Umlauten umgehen und ist bei den meisten Operationen erträglich fix. Ferner besitzt er eine Art „Programmiersprache", mit der sich Editoranweisungen einfach wiederholen lassen. Leider ist die Bedienung des ED für erfahrene Anwender nicht einfach, weil viele Tasten ungewohnt belegt sind. Ein Beispiel: Die „Insert"-Taste fügt bei allen anderen Editoren eine Zeile **vor** der Cursorzeile ein – der ED fügt sie **danach** ein. Der Editor der neuen Version von Lattice-C erlaubt es, mehrere Dateien gleichzeitig zu verarbeiten; er wird zum Teil mit der Maus gesteuert.
Der Megamax-Compiler wird als einziger mit einem mausgesteuerten Editor ausgeliefert. Dieser kann mehrere Texte gleichzeitig bearbeiten und hat einige für C sehr nützliche Fähigkeiten. Er kann ganze Blöcke einrücken, per Mausklick können Texte zwischen Klammern markiert, kopiert, verschoben oder gelöscht werden. Auch die Geschwindigkeit des Editors ist erträglich. Doch auch er hat Macken: Der Cursor läßt sich nur mit der Maus bewegen. Die Cursortasten bewirken lediglich ein Durchrollen des Textes. Diese höchst unangenehme Eigenschaft macht das Editieren oft zur Qual. Außerdem zeichnet er sich in der Version 1.0 durch einige sehr gut plazierte Fehler (immer da, wo der Schaden am größten sein kann) aus: Wenn z. B. beim Ersetzen der zu ersetzende Text im Ersatztext enthalten ist (Glas, Wasserglas), so hängt sich der Editor in einer Endlosschleife auf. Der bearbeitete Text ist dann unwiderruflich verloren.

Compiliergeschwindigkeit

Bei größeren Programmen ist die Zeit für die Compilierung eines Programms bedeutsam. Es ist nicht besonders angenehm, bei jedem entdeckten und korrigierten Fehler eine Ewigkeit auf die erneute Compilierung zu warten. Dieses Problem läßt sich durch die Modularisierung eines Programms zwar eingrenzen, jedoch nicht beseitigen. Als Test wurde das Programm XLISP, das aus insgesamt 23 Modulen besteht, probeweise vollständig mit allen C-Compilern übersetzt.
Der DRI-Compiler ist der behäbigste von allen. 15 Minuten und 13 Sekunden ist er mit der Übersetzung aller Module beschäftigt. Der Lattice-Compiler ist mit genau 9 Minuten schon wesentlich schneller. Mit Abstand am schnellsten ist allerdings der Megamax, der in sage und schreibe 2 Minuten und 38 Sekunden fertig wird. Dieser Unterschied ist hauptsächlich darin begründet, daß der DRI vier, der Lattice zwei und der Megamax sogar nur einen einzigen Durchlauf für die Compilierung benötigt.

Die inneren Werte

Eine Entwicklungsumgebung läßt sich vom Programmierer notfalls selbst gestalten. Shells, vollständige Make-Versionen und Editoren gibt es in großen Mengen auf dem Markt. Was den Wert eines C-Compiler-Pakets ausmacht, ist natürlich der Compiler selbst. Zur Beurteilung gibt es verschiedene Kriterien.
Zuerst wird der Sprachumfang untersucht. C ist auf zweierlei Weise standardisiert: Einerseits durch den Buchklassiker „The C Programming Language" (K&R) von Kernighan und Ritchie. In diesem Buch ist der ursprüngliche Sprachumfang exakt definiert. Der zweite Standard stammt vom ANSI. Er basiert auf dem von K&R beschriebenen Standard, bietet aber einige Erweiterungen. So lassen sich laut diesem Standard Aufzählungen (enumerations) definieren. Bildlich vorgestellt kann dann eine Farbe die Werte einer Aufzählung aus Grün, Rot, Blau oder Braun annehmen. Außerdem können kompliziertere Datenstrukturen (structures), die aus mehreren Einzeldaten bestehen, als ganzes verarbeitet werden. Die Namen von Einzeldaten innerhalb von Datenstrukturen dürfen mehrfach verwen-

Allgemeines zu C-Programmen

Wer Basic kennt, weiß zwar, wie ein Programm funktioniert. Die Struktur eines C-Programms ist von der eines Basic-Programms grundsätzlich verschieden.
Ein C-Programm besteht auf oberster Ebene aus Modulen, in denen einzelne Programmteile enthalten sind (z. B. „Eingabe", „Berechnung", „Ausgabe"). Im einfachsten Fall besteht es aus nur einem Modul. Jedes Modul besteht aus Funktionen, von denen jede eine kleine Teilaufgabe des Programms übernimmt. („Zeichenkette ausgeben", „Bildschirm löschen" usw.) Funktionen können andere Funktionen aufrufen. (Die Funktion zum Ausgeben von zehn Zeichenketten ruft zehnmal die Funktion zum Ausgeben einer Zeichenkette auf.) Dabei können Daten an die aufgerufene Funktion übergeben werden (die auszugebende Zeichenkette), die Argumente bzw. Parameter. Eine Funktion kann an die aufrufende Funktion einen Wert zurückgeben. (So gibt z. B. die Funktion für die Berechnung des Sinus den Sinus des übergebenen Arguments zurück.)

det werden. Die wichtigste Neuerung ist das „function prototyping". Mit diesem Verfahren kann ein C-Compiler überprüfen, ob die Datentypen von übergebenen Werten stimmen, was die Erstellung großer Projekte sehr erleichtert.
Keiner der getesteten Compiler erfüllt den ANSI-Standard. Besonders schwer wiegt das allseitige Fehlen des so ungemein praktischen function prototyping. Der DRI-Compiler verarbeitet noch nicht einmal den K&R-Standard, der ein absolutes Minimum darstellt. So lassen sich nicht einmal innerhalb von Blöcken Variablen deklarieren. Keine der Erweiterungen des ANSI-Standards wird geboten. Zusätzlich hat er diverse Einschränkungen, was die verwendbaren Datentypen betrifft: Die bei Lattice und Megamax vorhandenen Typen unsigned char, unsigned long und double werden nicht unterstützt, aber immerhin automatisch durch char, long und float ersetzt.
Der Lattice-Compiler sieht da schon wesentlich besser aus. Über den vollen K&R-Standard hinaus lassen sich Aufzählungstypen verwenden, Datenstrukturen einander zuweisen und von Funktionen zurückgeben. Außerdem lassen sich Strukturkomponenten mehrfach verwenden. Lediglich die Übergabe von Strukturen und das „function prototyping" werden nicht unterstützt.
Beim Megamax-Compiler fehlt nur der Aufzählungstyp (und natürlich das function prototyping) des ANSI-Standards. Er bietet den größten Sprachumfang der getesteten Compiler.

Beeinflußbarkeit der Compiler

Die Compilierung eines Programms läßt sich unter Umständen beeinflussen. So könnte z. B. verlangt sein, längere Variablennamen zuzulassen, zusätzliche Informationen einzubinden usw.
Der Compiler von DRI beschränkt sich dabei im wesentlichen auf die Angabe der Dateien, die vom Compiler erzeugt werden. Er läßt sich auch dazu bewegen, das Produkt des Preprozessors auf dem Bildschirm auszugeben, mit oder ohne zusätzliche Informationen; das ist aber auch schon alles.
Beim Lattice-Compiler läßt sich über Optionen praktisch alles festlegen. Ob es sich um die Zulassung längerer Variablennamen, die Festlegung des benötigten Speicherplatzes, die Erzeugung von Preprozessorcode, die Definierung spezieller Symbole oder anderes handelt – hier liegt der Lattice-Compiler eindeutig weit vorn.
Beim Megamax-C gibt es keinerlei Optionen, was ein großer Nachteil ist. In der Runtime-Bibliothek, die als Quelltext beiliegt, können einige Werte geändert werden.

Der erzeugte Code

Wir kommen zum wichtigsten qualitätsbestimmenden Faktor eines C-Compilers: dem Code, den er erzeugt. Jeder Compiler geht bei der Übersetzung zwar grob auf die gleiche Weise vor, in den Feinheiten gibt es jedoch wichtige Unterschiede. Es kommt dabei vor allem auf die Kompaktheit und die Geschwindigkeit des Codes an. Zwar sind dies Faktoren, die beim schnellen Atari ST mit seinem großen Speicher längst nicht mehr so stark ins Gewicht fallen. Aber dennoch: bei komplexen, rechenintensiven Programmen ist der Programmierer über jedes Prozentchen Zeit glücklich, daß der Compiler ihm spart.
Zunächst zur Geschwindigkeit des Codes, die von verschiedenen Faktoren abhängig ist. Der erste ist die Art der Daten, mit denen gerechnet wird. Dazu ein Beispiel: Zwei fünfziffrige Zahlen lassen sich natürlich leichter und schneller addieren, als zwei Zahlen mit je zehn Ziffern. Ähnlich ist es im Computer, auch wenn dieser mit Binärzahlen rechnet. Zwei Zahlen mit 16 Binärstellen lassen sich schneller addieren als solche mit 32. Und genau da kommt ein gravierender Faktor zum Tragen: Der DRI- und der Megamax-Compiler rechnen normalerweise mit 16 Binärstellen (sie können auch mit 32), während der Lattice-Compiler immer mit 32 rechnet. Das liegt daran, daß in älteren C-Programmen Zeiger (die 32 Bit „breit" sind) wie ganze Zahlen behandelt werden. Der DRI und der Megamax schneiden dabei einfach 16 Bit ab, was zum Programmabsturz führt. Der Compiler von Lattice wird damit fertig. Allerdings muß er mit einer geringeren Rechengeschwindigkeit bei ganzen Zahlen dafür bezahlen.
Das nächste Kriterium ist die Geschwindigkeit, in der die Standardfunktionen realisiert werden. Es handelt sich um Funktionen zur Verarbeitung von Fließkommazahlen (Dezimalzahlen), zur Ein- und Ausgabe auf Diskette, zur Verarbeitung von 32-Bit-Zahlen, zur Verarbeitung von Zeichenketten usw. Hinzu kommen die Optimierungen, die ein Compiler durchführen kann, wenn redundante oder zu vereinfachende Anweisungen im C-Programm auftreten. (So ist z. B. ((a*1)+0)/1) redundant, weil sowieso nur a herauskommt.) Auch lassen sich bei komplizierteren Operationen Teile, die mehrfach vorkommen, nur einmal berechnen, (z. B. ((a/10)+(a/10))/(a/10)). Ferner lassen sich Multiplikationen und Divisionen mit Zweierpotenzen durch sehr effiziente Maschinencodeanweisungen ersetzen.
Um diese Eigenschaften zu testen, werden spezielle Programme eingesetzt (Benchmarks), die je eine Eigenschaft des Compilers ansprechen. Benchmarks haben in der Regel Ablaufzeiten im Minutenbereich, um eine Zeitmessung vornehmen zu können und um Unterschiede zwischen den Compilern herauszustellen. Für diesen Testbericht wurden 12 Benchmarks geschrieben, die auf unterschiedliche Aspekte eingehen.
Als erstes ist die Fließkommaarithmetik an der Reihe. Hier gibt es zwei Benchmarks, von denen der erste auf die „Grundrechenarten", der zweite auf die mathematischen Funktionen wie Sinus, Cosinus, Logarithmus usw. eingeht.
Dann kommt ein Klassiker unter den Benchmarks: Das Sieb des Erathostenes, mit dem Primzahlen ermittelt werden. Er darf bei keinem Compiler-Test fehlen. Für diesen Test wurde er in einer stark vereinfachten Form realisiert.
Als nächstes ist die Geschwindigkeit des Diskettenzugriffs dran. Im Benchmark wird eine 64 KByte große Datei erst zeichenweise geschrieben und dann ebenso wieder gelesen.
Zwei Programme für die Verarbeitung von 32-Bit-Zahlen bilden den nächsten Benchmark. Das zweite Programm ist mit dem ersten bis auf die Besonderheit identisch, daß die wichtigsten Variablen in Prozessor-Registern gehalten werden, mit denen sehr viel schneller gerechnet werden kann.
Ein Test mit Zeigern wurde auf die gleiche Weise durchgeführt: einmal mit und einmal ohne Prozessor-Register.

Alsdann mußten sich die Zeichenkettenfunktionen einem Härtetest unterziehen, wobei Zeichenketten tausendfach zusammengesetzt, verglichen, bemessen, kopiert und auseinandergerupft wurden. Der letzte Benchmark testet die Optimierungsmaßnahmen des Compilers.

Drei Besonderheiten sind zu den Ergebnissen in *Bild 1* anzumerken: Die drei beim DRI-Compiler mit einem Strich gekennzeichneten Programme stürzten ab. Bei den Optimierungen ist zu sagen, daß beim Megamax keinerlei Optimierungen vorgenommen wurden, auch wenn die Dokumentation dies behauptet. Der Lattice ist da schlauer. Er kann Multiplikationen und Divisionen mit Zweierpotenzen ersetzen und behält Zwischenergebnisse im „Gedächtnis". Auch wenn der Benchmark beim DRI nicht lief: Der DRI kann Multiplikationen ersetzen. Zwischenergebnisse behält er nicht. Dafür kann er aber redundante Ausdrücke optimieren. Die Fließkomma-Benchmarks zeigen geradezu unglaublich gute Ergebnisse für den DRI. Das liegt daran, daß er sehr ungenau und damit schneller rechnet. Wir gehen darauf noch gesondert ein.

Wenn die Zeiten addiert werden (beim DRI-Compiler wurden die fehlenden Zeiten durch „pessimistische Schätzungen" ersetzt), so landet der DRI mit 21:52,9 Minuten knapp vor dem Megamax mit 22:46,1 und dem Lattice mit 28:02,8. Allerdings würde der Megamax bei Wegfall der Fließkomma-Benchmarks den DRI überholen. Insgesamt sind die Geschwindigkeitsunterschiede zwischen den einzelnen Compilern so gering, daß sie nicht das Hauptkriterium für die Bewertung darstellen können.

Der Megamax hat einen Vorteil dadurch, daß er eine spezielle Adressierungsart im Code verwendet, die schnellere und kürzere Programme erzeugt. Dafür können zusammenhängende Code- und Datensegmente stets nur 32 KByte groß sein. Bei größeren Projekten müssen spezielle Anweisungen eingefügt werden, was jedoch nicht weiter problematisch ist.

Benchmark	DRI	Lattice	Megamax
Fließkomma (Grundrechenarten)	6,6	38,0	34,3
Fließkomma (math. Funktionen)	3,4	01:29,9	03:32,0
Sieb des Erathostenes	02:18,0	03:04,1	02:08,3
Disketten-Schreibzugriff	02:38,0	01:21,1	60,0
Disketten-Lesezugriff	01:48,7	32,5	30,3
32-Bit-Zahlen	02:20,9	04:52,4	02:41,5
32-Bit-Zahlen in Registern	01:53,7	04:37,3	02:13,8
Zeiger	–	04:07,7	04:18,0
Zeiger in Registern	–	03:08,8	03:14,9
Zeichenketten	01:13,3	02:48,2	01:02,7
Optimierungen	–	01:22,8	01:30,3

Bild 1. Ergebnisse der drei C-Compiler

	DRI	Lattice 3.04	Megamax
Lieferumfang (außer Selbstverständlichkeiten)	Assembler, Librarian, RCS[1]), Debugger, Kommandoprozessor	GEM-Bibliothek-Quelltexte Shell, Debugger, RCS[1])	Assembler, Librarian, RCS[1]), Disassembler, Shell
Entwicklungsumgebung	Kommandoshell	Shell MENU+	Graphic Shell mit Make
Sprachumfang	keine Blockvariablen	Struktur-zuweisung, –rückgabe, Aufzählungen	Struktur-zuweisung, –rückgabe, –übergabe
Besonderheiten Datentypen	kein unsigned char kein unsigned long kein double	int 32 Bit breit	
Optionen	wenig	sehr viele	keine
Compiliergeschwindigkeit	gering	ca. 2× DRI	ca. 3× Lattice
Fließkommagenauigkeit	sehr gering	erträglich	hoch
Assemblereinbindung	direkt	nur indirekt	direkt
Optimierungen	Redundanzen, Multiplikationen	Zwischenergeb., Mult., Div.	keine
Fehlermeldungen	umfangreich	sehr umfangreich	wenig
Linkgeschwindigkeit	gering	hoch (2× DRI)	hoch
Linker-Optionen	viele	sehr viele	keine
Bibliotheksumfang	Standard	viele Unix-Funk.	Standard
Librarian	ja	nein	ja
GEM-Unterstützung	ja	ineffizient	ja
Dokumentation	umfangreich, aber fehlerhaft	umfangreich, ohne GEM	umfangreich, teilweise knapp
Debugger-Unterstützung	ja	ja	nein
Debugger	ja, SID	ja	nein

[1]) RCS: Resource Construction Set

Bild 2. Drei C-Compiler für den Atari ST im Vergleich

Der Umfang des Codes

Der Umfang des compilierten Codes läßt sich nur an größeren Programmen halbwegs vernünftig vergleichen. Als Test wurde hier nochmals XLISP verwendet. Dabei zeigten sich kaum Unterschiede. Alle Compiler erzeugten etwa 55 KByte. Zwar ließ sich beim DRI kein lauffähiges Programm produzieren, an der Objektcodegröße ließ sich dies jedoch abschätzen.

Für mathematische Anwendungen kann es u. U. von Bedeutung sein, wie genau Programme mit Fließkommazahlen umgehen können. Die Unterschiede zwischen den Compilern scheinen da ziemlich gravierend zu sein. Im Testprogramm wurde die Zahl 1000 x-mal durch 6 geteilt und anschließend wieder x-mal mit 6 multipliziert. Eigentlich müßte dann wieder 1000 herauskommen. Nach x Durchläufen ergab sich eine Abweichung von mehr als 1. Schon bei x=29 verzweifelt hier der DRI. Er rechnet mit Abstand am ungenauesten, aber auch mit großem Abstand am schnellsten. Beim Lattice-C ist bei x=480 Schluß, beim Megamax-C gar erst bei x=1984. Dies ist natürlich nur ein punktueller Test. Da aber „komplexere" Funktionen auch Multiplikationen und Divisionen benutzen, hat sich in der Praxis gezeigt, daß die Werte halbwegs Allgemeingültigkeit besitzen.

Fehlermeldungen

Wenn in einem C-Programm syntaktische Fehler enthalten sind, muß der Compiler darüber Auskunft geben. Er muß anzeigen, wo sich der Fehler befindet und von welcher Art der Fehler ist. C-Compiler sind da unterschiedlich pingelig und genau.

Compilierung eines C-Programms

Damit ein C-Programm ausgeführt werden kann, bedarf es einer Übersetzung in Maschinencode. Diese Übersetzung übernimmt der Compiler. Die Übersetzung eines C-Programms gestaltet sich in mehreren Teilen:
Zunächst erzeugt der Compiler aus dem eingegebenen Quelltext den Objektcode, der die Maschinencodeanweisungen enthält. Dieser Objektcode ist ein Modul. Dabei kann ein Compiler aus mehreren Schritten, den Läufen oder Passes bestehen, die jeweils nur einen Teil der Compilierung übernehmen.
Zu den (dem) Objektcode(s) müssen nun noch ein Startmodul und einige Module aus Bibliotheken hinzugefügt werden. Eine Bibliothek ist eine Sammlung von Objektcodes. Dieses Zusammenbinden übernimmt der Linker, der Vorgang wird in der Regel Linken genannt.

Der DRI-Compiler hat sehr umfangreiche Fehlermeldungen, was Fehler im Quelltext anbelangt. Wenn es allerdings Probleme mit der Disketten-Ein- und -Ausgabe gibt, spielt er oft verrückt. Wenn er eine Datei nicht lesen oder schreiben kann, merkt er das häufig gar nicht und „compiliert" lustig weiter. Häufig befindet sich dann hinterher „Müll" auf der Diskette.
Beim Lattice-C finden sich die meisten Fehlermeldungen. Er meldet jede Stelle, bei der er einen Fehler auch nur vermutet. Er steht da dem Unix-Lint in fast nichts nach. Das ist gut, weil auch nicht offensichtliche Fehler erkannt werden. Auf der anderen Seite ist es aber auch häufig unangenehm, weil der Compiler in fast jedem Programm Fehler sieht.
Beim Megamax-C sieht es mager aus. Es gibt nur wenige Fehlermeldungen, die zudem nur selten den „Punkt" treffen. Meistens ist der Fehler im Programm ganz anderer Natur, als es der Compiler angibt. Nach einem Fehler „verschluckt" er sich nämlich häufig und produziert nur noch Unsinn. Dieses Manko ist allerdings nicht so schwerwiegend, da „echte Programmierer" die Art eines Fehlers ohnehin eher selbst herausfinden, als dies dem Compiler zu überlassen.
Größere Programme sind naturgemäß fehlerhaft. Auch C-Compiler sind davon nicht unberührt. Gerade hier sind Fehler besonders unangenehm, weil der Programmierer Fehler in der Regel in seinem eigenen Programm sucht. Stecken sie im Compiler, sucht er natürlich vergeblich. Hier haben sich die Compilerbauer wirklich Mühe gegeben. Lediglich der DRI-Compiler ist negativ aufgefallen. Er hat Schwierigkeiten bei Berechnungen von Fließkommazahlen, wo er sich manchmal irrt.

Der Linker

Nicht allein der Compiler ist für das resultierende Programm entscheidend. Auch der Linker, der ein compiliertes C-Programm ablauffähig macht, hat eine entscheidende Bedeutung. Beim DRI-Compiler wird der Standard-Linker LINK68 mitgeliefert. Er glänzt durch eine große Beeinflußbarkeit durch die Angabe von Optionen in der Kommandozeile. Allerdings ist er relativ langsam. Für das Linken des Compilates von XLISP benötigt er etwas über 2 Minuten.
Der Lattice-Compiler verfügt über einen leistungsfähigen Linker. Er wird über eine Steuerdatei über seine Aufgaben informiert. Für XLISP benötigt er 38 Sekunden. Der Megamax-Linker läßt sich dagegen kaum beeinflussen. Man kann ihm lediglich sagen, welche Module und Bibliotheken er zusammenzubinden hat. Eine Beeinflussung durch Optionen ist nicht vorgesehen, dafür ist er aber recht schnell. XLISP hat er in 48 Sekunden ablauffertig bereitgestellt.

Die Bibliotheken

Die Bibliotheken liefern ein weiteres wichtiges Beurteilungskriterium für C-Compiler. In ihnen sind viele nützliche Funktionen enthalten, die dem Programmierer Arbeit abnehmen. Diese Funktionen werden ebenfalls von K&R standardisiert. Viele Compilerbauer liefern aber Erweiterungen mit. Je umfangreicher die mitgelieferten Bibliotheken sind, desto mehr Arbeit wird dem Programmierer erspart. Ein weiteres Kriterium ist die Fehlerfreiheit der bereitgestellten Routinen.
Der DRI-Compiler bietet eine Bibliothek, die ziemlich streng an K&R angelehnt ist. Nur wenige Erweiterungen sind programmiert. Auszusetzen ist die große Fehlerhaftigkeit dieser anscheinend „auf die Schnelle" gestrickten Funktionen. In kaum einem Bereich gibt es nicht mindestens eine Funktion, die einen Fehler besitzt – ob es sich um die Ein- und Ausgabe oder die Speicherverwaltung handelt. Dies ist auch der allergrößte Mangel am DRI, was ihn stark abwertet.
Der Compiler von Lattice bietet mit über 30 Funktionen die mit Abstand umfangreichste Bibliothek an. Sie bietet neben K&R viele Unix-Funktionen, wenn auch nicht alle. Die Lattice-Bibliothek zeichnet sich außerdem durch eine sehr große Fehlerfreiheit aus.
Zum Megamax-Compiler gehört eine Minimalbibliothek, die fast keine über K&R hinausgehende Funktion bietet. Sie ist kompakt, schnell und relativ fehlerfrei.

Dokumentation

Ein Punkt, der oft überbewertet wird, ist die zu einem Compiler gelieferte Dokumentation. Sie stellt meistens nur eine Beschreibung der Fähigkeiten des Compilers dar. Ein Freak wird diese Fähigkeiten aber lieber selbst ermitteln. Dokumentationen ist häufig nicht zu trauen. Dennoch soll der Papierkrieg nicht unerwähnt bleiben. Übrigens ist alles in englisch.
Zum Entwicklungssystem von DRI gehört die komplette Systemdokumentation zum Atari ST. Von den untersten Systemaufrufen über das Betriebssystem zum GEM ist alles beschrieben. Auch die Systemtools und der C-Compiler sind mit „stapelweise" Papier versehen. Doch ist viel darunter, was auf den Atari ST nicht ganz oder gar nicht zutrifft. So ist z. B. die GEM-Dokumentation ohne jegliche Änderung von der des IBM-PC kopiert worden. Die Dokumentation zum C-Compiler wurde ebenfalls übernommen, vom CP/M-68K nämlich, von dem er ursprünglich stammt. So paßt denn vieles auch nicht auf die Atari-Version. Es wird behauptet, daß der DRI keine Fließkommazahlen verarbeiten kann, viele erlaubte Compileroptionen bedürfen der Erläuterung. Wenn davon abgesehen wird, handelt es sich allerdings um präzisen Text, eigentlich genau das, was sich jeder Programmierer wünscht. Nur: Was hilft das dann noch? Sehr praktisch ist der „C Style Guide", der darauf eingeht, wie sich C-Programme strukturieren lassen, wie sie portabel und lesbar gemacht werden.
Die Lattice-Dokumentation läßt kaum Wünsche offen. Der Compiler und der Linker mit ihrer Funktion und den Optionen wird ausführlich beschrieben. Die Lattice-Portierung wird genau beschrieben, und jeder Bibliotheksfunktion ist eine eigene Seite mit einer übersichtlichen und genauen Erklärung eingeräumt. Leider sind die GEM-Funktionen jeweils nur mit einer einzelnen Zeile der Erläuterung versehen. Auch die GEMDOS-Funktionen sind nicht gesondert erwähnt. So ist der GEM-Einsteiger gezwungen, sich Sekundärliteratur zu besorgen. Von Philgerma aus München wird für 28 DM eine sehr gute, deutsche Dokumentation zum Lattice-C angeboten (600 Seiten). Die Dokumentation zum Megamax glänzt ebenfalls durch ihre Beschreibung der Bibliothek in knapper und präziser Weise. Auch die GEM- und GEMDOS-Funktionen kommen nicht zu kurz. Zu knapp da-

vongekommen sind die Beschreibungen der einzelnen Programme. Sehr unangenehm ist z. B., daß die etwas unübliche Syntax des eingebauten Assemblers nur in der nicht jedem verständlichen EBNF angegeben ist. Die Beschreibung des Resource Construction Set setzt eigentlich voraus, daß der Leser schon GEM-Profi ist. Doch wer bereit ist, ein wenig zu probieren, wird auch damit zufrieden sein. Schwerer wiegt das Fehlen von Erläuterungen zu den einzelnen Fehlermeldungen, die beim DRI und beim Lattice vorhanden sind.

Einbindung von Assembler-Programmen

Manchmal ist es erforderlich, einzelne Funktionen eines C-Programms in Assembler zu schreiben. Die Compiler unterstützen dies auf unterschiedliche Weise. Da der DRI als einen der Compilerläufe einen Assembler hat, lassen sich bei ihm direkt Assemblerbefehle in das C-Programm einbinden. Außerdem können Assembler-Module getrennt assembliert und hinterher dazugelinkt werden.
Einen weiteren Vorteil erwirbt sich der DRI-Compiler hier: Dadurch, daß er Assemblerquelltext erzeugt, lassen sich nachträglich Optimierungen durchführen.
Der Lattice-Compiler sieht hier „alt" aus. Ein Assembler ist nicht im Lieferumfang vorhanden. Der Anwender muß sich einen separaten Assembler zulegen, mit dem er dann einzelne Module erstellen und zu C-Programmen dazulinken kann. Von-Hand-Optimierungen sind beim Lattice-C fast unmöglich.
Der Compiler von Megamax hat einen „eingebauten" Assembler, der die Einbindung von Maschinenbefehlen vereinfacht. Leider hat er einige Fähigkeiten eines „richtigen" Assemblers nicht. So lassen sich keine Daten direkt ablegen. Durch das unübliche Objektcodeformat des Megamax-C können bis jetzt auch noch keine anderen Assembler benutzt werden. Allerdings lassen sich die Objektcodes mit einem mitgelieferten Programm disassemblieren und (mit einigem Aufwand) später wieder assemblieren.

Debugging

Wenn in einem Programm Fehler stecken, lassen sie sich nicht immer durch Überprüfen des Quelltextes finden und beseitigen. Dann wird es notwendig, das Programm in Einzelschritten zu verfolgen und den Fehler zu lokalisieren. Dies geschieht mit einem speziellen Programm, dem Debugger. Bisher sind auf dem Atari ST lediglich Maschinencode-Debugger verfügbar. Es muß also der vom Compiler erzeugte Code analysiert werden, um dann auf Fehler im Quelltext zu schließen. Auf Rechnern wie dem IBM-PC gibt es schon Quelltext-Debugger, die ein Betrachten und schrittweises Ausführen des Quelltextes ermöglichen, was die Sache sehr erleichtert. Auf dem Atari ST gibt es aber immerhin schon symbolische Debugger, die es erlauben, ungefähr die Stelle im Quelltext zu lokalisieren, zu der ein Maschinenbefehl gehört. Diese Fähigkeit muß der C-Compiler unterstützen.
Der DRI-Compiler kann Informationen für einen symbolischen Debugger einbinden. Ein solcher ist auch im Entwicklungssystem enthalten, der SID nämlich. Dieser (recht einfache) Debugger ermöglicht dann die Einzelschrittausführung eines Programms

Lexikon

Begriff	Erklärung
ANSI	– American National Standard Institute Amerikanisches Normungsinstitut (ähnlich DIN)
Assembler	– a) Sprache, die direkt in –> Maschinencode übersetzt wird b) Übersetzungsprogramm von Assembler nach Maschinencode
Batch-Prozessor	– Programm, das Befehle automatisch ausführt
Betriebssystem	– –> Bibliothek mit häufig benutzten Funktionen
Bug	– Fehler im Programm
Benchmark	– Testprogramm
Bibliothek	– Sammlung von –> Objektcodes
char	– –> Datentyp für Buchstaben
Compiler	– Übersetzungsprogramm für –> Hochsprachen
CP/M-68k	– –> Betriebssystem
Datenstruktur	– Zusammensetzung mehrerer Daten
Datentyp	– Art eines Datums, z. B. ob es sich um eine ganze Zahl, eine Dezimalzahl oder einen Buchstaben handelt
Debugger	– Programm zum Finden von Fehlern in anderen Programmen
Disassembler	– Programm zum Übersetzen von –> Maschinencode in –> Assembler
double	– –> Datentyp für doppelt genaue –> Fließkommazahlen
DRI	– Digital Research, Inc., Softwarefirma
EBNF	– Erweiterte Backus-Naur-Form, Darstellungsform für Syntax
Editor	– Programm zum Erstellen und Ändern von Texten
Festplatte	– Nicht auswechselbare Magnetplatte mit hoher Zugriffsgeschwindigkeit und großer Kapazität
Fließkommazahl	– Dezimalzahl (mit Stellen nach dem Komma)
float	– Datentyp für einfach genaue –> Fließkommazahlen
GEM	– Graphics Environment Manager, –> Bibliothek von Grafikfunktionen
int	– Datentyp für ganze Zahlen
K&R	– Kernighan & Ritchie
Hochsprache	– Computersprache, die stärker an die menschlichen Denkstrukturen angepaßt ist als –> Assembler
Librarian	– Programm zur Verwaltung von Bibliotheken
Linker	– Programm zum Verbinden mehrerer Objektmodule zu einem Programm
Maschinencode	– direkt vom Mikroprozessor ausführbarer Code
Modul	– Teil eines Programms
Objektcode	– compiliertes Programm
Option	– mögliche Beeinflussung eines Programms
Pass	– Durchlauf eines Compilers
Preprozessor	– Teil des C-Compilers
Prozessorregister	– Speicher, der besonders schnell angesprochen werden kann
Quelltext	– ursprüngliches Programm
RAM-Disk	– simuliertes Diskettenlaufwerk im Arbeitsspeicher
Resource Construction Set (RCS)	– Programm zum Erstellen von Bildschirmelementen
Runtime	– Modul, das zum Ablauf eines Programms notwendig ist
Shell	– Programm zur Steuerung anderer Programme
Struktur	– –> Datenstruktur
Tool	– Werkzeug, Hilfsmittel zum Programmieren
Unix	– –> Betriebssystem, in C geschrieben
unsigned char	– Datentyp für Buchstaben
unsigned long	– Datentyp für 32-Bit-Zahlen
XLISP	– Public Domain Lisp Interpreter
Zeiger	– Verweis auf Daten

und kann so die Fehlersuche sehr erleichtern. Beim Lattice-C können Debugger-Informationen eingebunden werden, allerdings wird mit der neuen Version ein sehr guter Debugger mitgeliefert. Der Megamax sieht hier nicht besonders gut aus. Weder unterstützt er einen Debugger, noch wird einer mitgeliefert.

GEM-Unterstützung

Alle C-Compiler bieten Bibliotheken mit den GEM-Funktionen. Der Compiler von Lattice hat hier allerdings seine Probleme, weil er mit den schon oben erwähnten 32-Bit-Zahlen rechnet. Die GEM-Funktionen erwarten 16-Bit-Zahlen. So muß die GEM-Bibliothek selbständig für die Umsetzung sorgen, was sie unnötig umfangreich macht.
Außerdem werden bei allen hier behandelten C-Compilern Programme mitgeliefert, die das Erstellen der für GEM so wichtigen Resource-Dateien ermöglichen, was für den GEM-Programmierer unverzichtbar ist.
Der professionelle Programmierer wird sich in der Regel eigene Bibliotheken zusammenstellen, z. B. wenn es um Bildschirmfunktionen oder Grafik geht. Für die Erstellung von Bibliotheken wird ein spezielles Programm, ein Librarian benötigt. Derartige Programme werden nur beim DRI- und beim Megamax-Compiler mitgeliefert.
Bei keinem C-Compiler werden die Quelltexte zu der mitgelieferten Bibliothek bereitgestellt. Vor allen Dingen beim DRI wäre das unverzichtbar, weil dann die Fehler in den Bibliotheken nachträglich ausgemerzt werden könnten. Beim DRI und beim Megamax werden lediglich die Runtime-Quelltexte mitgeliefert. Immerhin werden beim Lattice die Assembler-Quelltexte zu den GEM-Bibliotheken mitgeliefert. Doch sie sind letztlich für den Programmierer auch weniger interessant.

Fazit

„Welcher ist denn nun der beste?", sollte die Abschlußfrage lauten, „Welche ‚Note' bekommt der DRI, der Lattice, der Megamax?". Dies läßt sich leider nicht mit der Bestimmtheit eines Standard-Stiftung-Warentest-Tests sagen, wie dies halt allgemein bei Computern und Software so ist. Die Frage muß richtiger lauten „Welcher ist der richtige für wen?" Darauf läßt sich schon eher eingehen:
Der DRI-Compiler hat viele Nachteile: Er ist langsam, behäbig, platzfressend, fehlerhaft und ungenau. Der hohe Preis mag außerdem abschrecken. Er hat aber auch Vorteile: Er ist flexibel, optimierbar, erzeugt kompakten Code, hat eine sehr schnelle Fließkommaarithmetik. Außerdem ist sein Lieferumfang nach wie vor unüberboten. Der DRI ist sicherlich nichts für C-Einsteiger. Für den Profi stellt er gerade aufgrund seiner Vorteile eine Herausforderung dar. Durch seine große Flexibilität ist er der Compiler, der den größten Freiraum bei geeigneter Programmierung offenläßt.
Beim Lattice fallen die 32-Bit-Zahlen und die ineffiziente GEM-Unterstützung negativ ins Gewicht. Dafür ist er weitgehend fehlerfrei, besitzt eine umfangreiche Bibliothek und ist weitestgehend standardisiert. (Den Lattice gibt es für viele andere Rechner.) Er ist für Einsteiger das richtige. Außerdem ist er für Umsetzungen von Fremdprogrammen interessant.
Die Hauptnachteile des Megamax-C sind seine Fehlerbehandlung, seine fehlenden Optimierungen und seine kleine Bibliothek. Dafür bietet er eine gute Entwicklungsumgebung, einen sehr schnellen Compiler, einen Assembler und einen großen Lieferumfang. Er unterstützt die professionelle Software-Entwicklung am besten. Jetzt stehen Sie vor der Wahl.

Neues zur Büroautomation von Data General

Mit der Ankündigung neuer und erweiterter Produkte zur Büroautomation trat Data General (DG) Ende Januar in München an die Öffentlichkeit. Das Büroautomationssystem CEO wurde in der Version 3.00 hinsichtlich Textverarbeitung, Dokumentverarbeitung und Electronic Mail verbessert; der CEO Desktop Composer ist eine Anpassung des Xerox Ventura Publishers an die Data-General-Systeme; DG/PC*I steht für die hard- und softwaremäßige Unterstützung der Integration von PCs in Netze zusammen mit Abteilungs- und Mainframe-Systemen.
Seit der Auslieferung der ersten CEO-Versionen im November 1981 sind ca. 600 Mannjahre in dieses Produkt investiert worden, mittlerweile existieren Versionen in 16 verschiedenen Landessprachen. Damit hat, nach Meinung des Produkt-Managers für Büroautomation Wolf Dawson, CEO eine gewisse Reifephase erreicht, in der total neue Funktionen jetzt nicht mehr zu erwarten sind.
CEO bietet jetzt unter anderem komfortable Textverarbeitungsfunktionen (z.B. Formularverarbeitung), erweiterte Druckfunktionen (z. B. Unterstützung postscriptfähiger Drucker), ein globales Netzverzeichnissystem, Versenden ganzer Aktenordner als Einheit, Angabe des Zustellungszeitpunktes einer elektronischen Post-Nachricht, mehr Zeilen für Kurznachrichten, automatische Archivierung, Restaurierung gelöschter Dateien im elektronischen Posteingangsfach eines Benutzers, eine Post-Option, bei der entweder nur gelesen, gelöscht oder beantwortet werden kann, Rechtschreibungsprüfung an beliebigen Stellen in CEO, Terminplanung durch CEO-Unterfunktion, gleichzeitige Überwachung von vier verschiedenen Kalendern, netzübergreifenden Zugriff auf Kalender.
Der CEO Desktop Composer nutzt die Kommunikationsmöglichkeiten von CEO und der Netzwerksoftware DG/PC*I; die Benutzer des CEO Desktop Composer können Informationen untereinander, mit den Abteilungsrechnern der Eclipse MV-Familie und mit Mainframes teilen und gemeinsam nutzen. Ein Texterfassungssystem ist CEO Desktop Composer nicht, Texte können von CEOtext, aber auch von WordPerfect, Wordstar, MS Word, Xerox Writer oder als normale ASCII-Dateien z.B. von Lotus 1-2-3 oder dBase III übernommen werden. Unter Unix wird es CEO innerhalb der nächsten zwei Jahre noch nicht geben, da es eng verknüpft ist mit dem Data-General-Betriebssystem.
Im Focus auf Zielmärkte und Marktsegmente sieht Karl Weißkopf, Geschäftsführender Direktor von Data General, die Zukunft für sein Unternehmen. Angesprochen werden sollen vor allem Finanzinstitute, die chemische Industrie (insbesondere im Pharma- und Petro-Bereich) sowie Behörden. Eine Folge dieses Konzeptes ist, daß DG in diesem Jahr auf keiner Universalmesse (auch nicht auf der CeBit) vertreten sein wird.
In der Kooperation mit Mannesmann-Kienzle seit gut einem halben Jahr sind bereits vier größere Projekte zustande gekommen, darunter auch mit der Münchner Messegesellschaft. Es gibt also immer noch Marktnischen, die in aller Stille von kompetenten Firmen abgedeckt werden.

Jürgen Plate

Super-Tabellenkalkulation

Quattro von Borland

Mit Reflex und Sidekick hat sich Borland schon früher an die Anwender von Software für den kaufmännischen Bereich gewendet. Waren diese Programme noch recht speziell, so zielt Borlands neustes Produkt Quattro direkt auf die Anwender von Tabellenkalkulationsprogrammen.

In Amerika, wo man in der Werbung nicht gerade zimperlich ist, wird Quattro direkt mit Lotus 1-2-3 verglichen und provokativ eine neue Zählweise eingeführt: „Eins-zwei-drei-Quattro". Beim Test stellte sich heraus, daß sogar die von uns getestete Vorab-Version so umfangreich ist, daß man kaum alle Features des Programms beschreiben kann, ohne eine langweilige, mehrseitige Tabelle zu produzieren (im Handbuch sind allein die Rechenfunktionen auf 75 Seiten beschrieben). Wir haben uns daher entschieden nur eine kurze Zusammenfassung zu geben.

Nachdem man die vier Quattro-Disketten auf die Festplatte (nach gewohnter Borland-Philosophie ohne Kopierschutz) kopiert hat, kann man sofort die ersten Schritte versuchen. Nach dem Starten des Programms ist außer einem leeren Kalkulationsblatt auf dem Bildschirm nichts zu sehen. Die einzige Taste, die man sich zunächst merken muß, ist der Schrägstrich. Mit dieser Taste kommt man in das Menü (*Bild 1*).

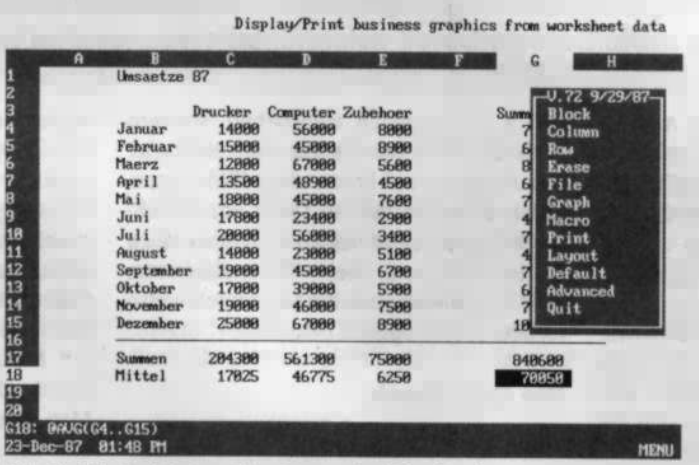


Bild 1. Mit der Taste „/" ruft man das Menü auf

Quattro unterscheidet selbständig zwischen einer Zahlen-, Formel- oder Stringeingabe, indem es sich am ersten Zeichen orientiert. Das kann manchmal bei der Eingabe von Strings schief gehen, wenn man eine Unterstreichungszeile bestehend aus Minuszeichen eingibt.

In solchen Fällen setzt man ein Apostroph an den Anfang des Strings, so daß man die Zeile über die Feldgrenzen hinweg eingeben kann. Alle Eingaben werden mit der Return- oder einer Cursor-Taste abgeschlossen und das Navigieren innerhalb des Blattes erfolgt mit den Tasten des Cursorblocks. Mit der Goto-Funktion (Taste F5) kann man schnell zu einem beliebigen Feld springen.

Während der Eingabe erscheinen die Daten in der Dialogzeile am oberen Bildschirmrand. Am unteren Bildschirmrand befindet sich die Statuszeile, in der neben Datum und Uhrzeit auch der Bearbeitungsstatus (z. B. „MENU" oder „EDIT") und die Fehlermeldungen angezeigt werden.

Kommandos und Menüs

Im Menü-Fenster wählt man die gewünschten Funktionen mit den Cursortasten oder durch Eingabe ihres ersten Buchstabens aus. Je nach Anforderung gelangt man so in weitere Untermenüs – narrensicher für den Einsteiger und den gelegentlichen Benutzer. Ist man mit dem Programm erst vertraut, kann man die Menüs den eigenen Bedürfnissen anpassen. Der Anwender braucht dazu lediglich „Abkürzungen" zu definieren. Ist man auf dem Menüpunkt, drückt man die Ctrl- und eine beliebige Buchstabentaste. Ab diesem Zeitpunkt ist der Menüpunkt mit der Ctrl- und dieser Buchstabentaste sofort erreichbar. Wenn das noch nicht reicht, kann man neue Untermenüs erzeugen, die Optionen einzelner Menüs tauschen oder sie umbenennen. Zu jedem Menüpunkt erscheint in der Dialogzeile eine kurze Erläuterung. Bei der Testversion von Quattro lief alles noch in Englisch ab und das Programm hatte auch Probleme mit den Umlauten – die deutsche Version ist aber schon in Arbeit.

Das Arbeitsblatt umfaßt maximal 8192 Zeilen mit je 256 Spalten. Die Länge einer Eingabezeile kann bis zu 240 Zeichen betragen – was auch für komplexe Formeln reichen sollte. Bei den Zahlenfeldern läßt sich das Format beliebig festlegen. Die Kopier- und Einfügefunktionen machen das Ändern von Kalkulationsblättern und die Datenerfassung leicht. Mit der Blockfüllfunktion kann man die Felder schnell mit einem Zeichen füllen. Hat man fortlaufende Werte (z. B. bei Zeitreihen), kann man ganze Blöcke füllen, indem man nur Anfangs- und Endwert eingibt. Der gelegentliche Anwender wird beim Definieren von Formeln wieder einen Blick ins Handbuch werfen müssen. Die Zeilen- und Spaltensummen werden z. B. durch „@SUM (G4..G15)" definiert. Die Mittelwerte bekommt man ebenso einfach über „@AVG(G4..G15)" (AVG: average). Es gibt Funktionen für fast alle denkbaren mathematischen Probleme: Grundrechenarten und Klammern, Trigonometrische Funktionen, Wurzeln, Exponenten und Logarithmen, Summe, Minimum, Maximum, Mittelwert, Standardabweichung. Mit zahlreichen Stringfunktionen lassen sich Texte manipulieren. Man kann auf die Zahl der Spalten oder Zeilen eines Blocks zugreifen oder Zellen indirekt adressieren („gib mir den Inhalt der Zelle, dessen Adresse in A9 steht"). Logische Verknüpfungen und Abfragen sind ebenso implementiert, wie finanzmathematische Operationen. Schließlich gibt es eine Reihe von Funktionen für die Suche und Auswahl im Datenbankteil.

Zellen, die sich ändern (aber nur diese und nicht das ganze Blatt) werden normalerweise automatisch berechnet, sobald ein Zahlenwert oder eine Formel geändert wird. Man kann die Automatik auch ausschalten und die Neuberechnung von Hand auslösen.

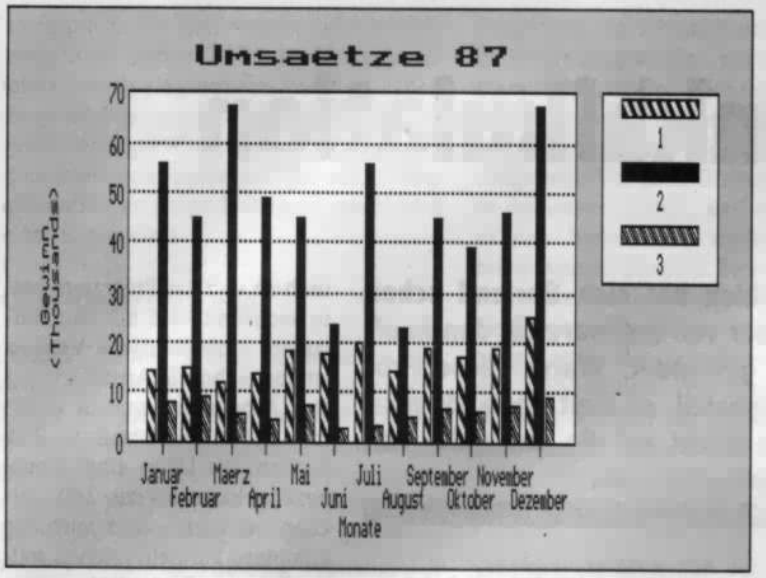


Bild 2. Eine Balkengrafik ist schnell erstellt

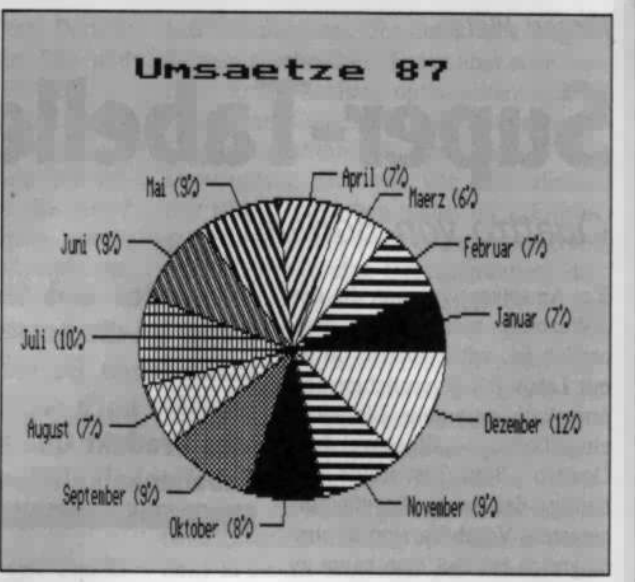


Bild 3. Tortengrafik mit prozentualer Darstellung

Man kann mit Quattro sogar zwei Arbeitsblätter bearbeiten (der Bildschirm wird dann horizontal oder vertikal geteilt) und von einem Blatt zum anderen kopieren. Um die Dateneingabe zu vereinfachen und sicherer zu machen, lassen sich Eingabemasken definieren und Felder sperren, die nicht ausgefüllt werden dürfen. Zusätzliche Hilfen bietet die Taste F1. Natürlich lassen sich auch alle Standardeinstellungen und Farben ändern. Auf Wunsch lassen sich die Daten auch je nach Wert einfärben – etwa zur Freude der Kaufleute die negativen Zahlen rot. Die Einstellungen lassen sich dauerhaft installieren. Auch merkt sich Quattro die zuletzt gewählten Einstellungen (z. B. das aktuelle Arbeitsblatt) für die folgende Sitzung. Aber nicht nur die Erscheinung des Programms, sondern auch die Druckausgabe läßt sich über die Randeinstellung, die Titel-, Kopf- und Fußzeilen und die Seitenlänge konfigurieren.

Geschäftsgrafiken inbegriffen

Einfache Grafiken kann man auch ohne tiefgehende Konsultationen des Handbuchs erzeugen (*Bild 2*). Es gibt insgesamt zehn verschiedene Grafiktypen. Für die Beschriftung werden mehrere Fonts mitgeliefert. Es lassen sich also sehr individuelle Grafiken erzeugen. In jedem Untermenü der Grafikfunktionen gibt es einen Punkt „View", mit dem man sich die erzeugten Grafiken schnell ansehen kann. Man kann mit den Parametern solange spielen, bis einem die Grafik zusagt. Auch die Kombination mehrerer Grafikarten (z. B. Balken- und Liniendiagramm) auf einem Bild ist möglich. Quattro unterstützt nicht nur die Farbgrafikkarte (bei der niedrigen Auflösung von 320 x 200 Punkten mehrfarbig), sondern auch die weit verbreitete Herculeskarte. Der Ausdruck auf Epson- oder IBM-kompatiblen Druckern lieferte sofort ein brauchbares Ergebnis; es lassen sich aber auch Plotter ansteuern. Beim Tortendiagramm *Bild 3* kann man einzelne Segmente herausziehen und so optisch betonen.

Die Datenbank

Als dritte Funktionsgruppe ist in Quattro eine Datenbank eingebaut. Wie bei den anderen Kalkulationsprogrammen werden die Datensätze zeilenweise abgelegt – Quattro kann 8191 Datensätze auf einmal speichern. Die Spalten des Arbeitsblatts kann man sich als die einzelnen Felder des Datensatzes vorstellen. So lassen sich bei der Datenbankanwendung die Spalten mit Überschriften versehen. Zur Auswahl der gewünschten Datensätze schreibt der Anwender die Suchbegriffe, verknüpft durch AND, OR und NOT in eine Zeile. In den Suchkriterien können Jokerzeichen, wie bei Dateinamen im DOS, verwendet werden. Das Programm stellt die gefundenen Datensätze invers dar. Man kann sie natürlich auch in einen freien Bereich des Arbeitsblatts kopieren. Die Datensätze lassen sich aufsteigend oder absteigend nach bis zu fünf Kriterien sortieren. Zur Weiterverarbeitung und Ausgabe der Daten bieten sich die Makrofunktionen von Quattro an.

Makros vereinfachen die Bedienung

Mit Makros, die eine Reihe von Tastenfunktionen ersetzen, kann man die Bedienung des Programms vereinfachen. Die Makros bieten jedoch einen Komfort, der schon an eine Programmiersprache erinnert. Es ist möglich, in den Programmablauf Schleifen, Verzweigungen und Unterprogramme aufzunehmen. Benutzereingaben zur Programmsteuerung können erfragt werden. Quattro erlaubt beliebig viele Makros. Damit man bei der „Programmentwicklung" mit Makros nicht ins Schleudern kommt, bietet Quattro umfangreiche Testhilfen. Man kann bis zu vier Breakpoints setzen und das Makro im Einzelschrittbetrieb ablaufen lassen. Während der Fehlersuche kann man sich den Inhalt von bis zu vier Zellen des Arbeitsblatts anzeigen lassen. Die erzeugten Makros sind zu Lotus 1-2-3 kompatibel und müssen nicht extra compiliert werden. Womit ich bei der Verträglichkeit von Quattro zu anderen Programmen wäre. Nach den Angaben des Handbuchs verarbeitet das Programm Dateien von Lotus, Reflex, dBase, Symphony sowie PFS- und DIF-Dateien. Auch reine ASCII-Dateien lassen sich importieren – ein spezielles Kommando splittet eine ASCII-Zeile in einzelne Felder. Übrigens kann man Quattro auch veranlassen alle Benutzeraktionen aufzuzeichnen; sei es zur Kontrolle oder um bei Fehlern zu einer früheren Version seines Arbeitsblatts zurückzukehren. Schließlich kann man von Quattro auch in DOS springen, dort normal arbeiten und mit EXIT wieder zurück ins Arbeitsblatt springen.

Quattro ist – nach der Eindeutschung durch den deutschen Vertreiber Heimsoeth, München – sicher für jene interessant, die bei einem Kalkulationsblatt nicht nur den attraktiven Preis, sondern auch die reichhaltigen Grafik- und Programmierungsmöglichkeiten zu würdigen wissen. □

Ulrich Rohde

Was man über Modula wissen sollte

Teil 2

Gewinnen Sie mit mc Überblick über Modula. Sie selbst können dann entscheiden, ob daraus eine Liebe wird oder Ablehnung. Modula hat etwas zu bieten, was keine andere Sprache so klar und praktikabel bietet: das modulare Konzept. Zunächst aber die normalen Spracheigenschaften.

Für alles Folgende sei die Sprachregelung getroffen, daß für Modula-2 stets nur noch Modula gesagt wird. Niklaus Wirth hatte zunächst eine auf Multi-Processing abzielende Sprache Modula entworfen. Im Zuge seiner Arbeiten an Betriebssystemen für Minicomputer entwickelte er dann aus Modula und Pascal die Sprache Modula-2, die das modulare Konzept von Modula „einer breiteren Öffentlichkeit zugänglich machte" und deshalb auch den Namen erbte. Daß mittlerweile eine weitere Sprache (Oberon) aus der Eidgenössischen Technischen Hochschule von Niklaus Wirth kommt, sei hier nur mitgeteilt.

Der erste Einstieg

Da es sich bei Modula um eine real existierende Sprache handelt, sind die im vorhergehenden Abschnitt geschilderten theoretischen Sachverhalte nicht pur verwirklicht, sondern soweit realisiert und auf eine Art realisiert, wie es eben Niklaus Wirth praktikabel erschienen. Basic-Programmierern muß gleich gestanden werden, daß Modula erstens keine Sprache ist, die interpretierend arbeitet. Und daß sie zweitens Eigenheiten besitzt, die einem Anfänger im Programmieren das Leben oft schwer machen, weil von ihm verlangt wird, Sprachdetails zu lernen, deren Sinn sich erst viel später in der Programmierpraxis zeigen. Nähert man sich der Sache aber ohne Vorurteil, dann wird man bald entdecken, daß Modula sehr logisch aufgebaut und ganz natürlich ist.

Die erste Tatsache erzwingt, daß nur fertige Programme getestet werden können. Die zweite, daß man lernen muß, mit einer strengen Überwachung durch den Compiler zu leben, der einem zunächst alle Ideen zu rauben scheint, obwohl man doch gerade in Modula angeblich so viel machen können sollte. Das ist aber in Pascal nicht viel anders und eine Seite von Modula kann mit Pascal verglichen werden. Es ist nicht falsch, wenn man sagt, daß Modula teilweise wie Pascal ist.

Wenn man mit dem Einstieg ganz unten anfängt, dann ist danach zu fragen, welche sprachlichen Elemente von Modula verwendet werden, um daraus Programme zu bilden. Niklaus Wirth führt dazu in seinem Buch als unterste „Elemente" der Sprache bestimmte Symbole ein, die nach den Regeln der Syntax aneinandergesetzt die wohlgeformten Programme ergeben. Diese Symbole sind in Klassen zusammengefaßt. Folgende Symbolklassen gibt es:

Namen, Zahlen, Strings, Operatoren, Begrenzer, Kommentare.

Die Modulasymbole werden aus den auf der Rechenmaschine verfügbaren Zeichen aufgebaut (repräsentiert).

Namen

(Identifier) sind Zeichenfolgen, bestehend aus Buchstaben und Ziffern, wobei das er-

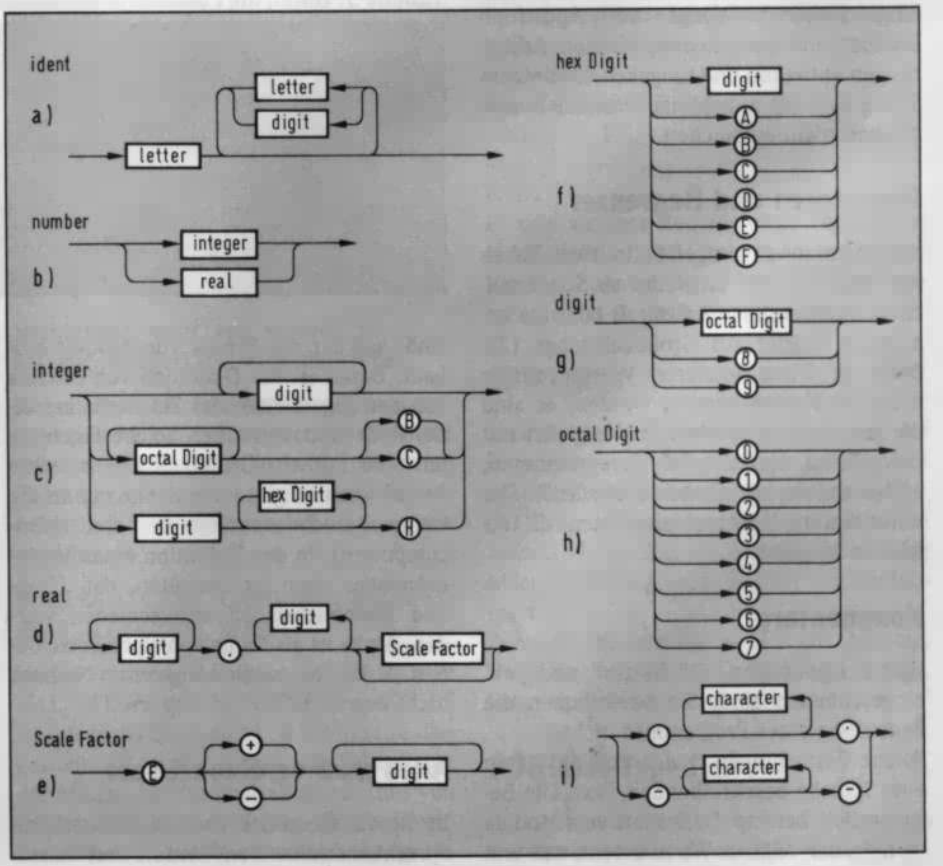


Bild 1. Die Syntaxdiagramme für den Aufbau der Zahlen

ste Zeichen ein Buchstabe sein muß. Beispiele: Hallo, xy, x567, dabei. Keine Namen: 3ßig, pascal_programm (Unterstrich ist Sonderzeichen, weder Buchstabe noch Ziffer).

Zahlen

sind entweder positive ganze Zahlen oder positive reelle Zahlen. Die positiven ganzen Zahlen sind aus Ziffern aufgebaut, sind Ziffernfolgen. Die positiven reellen Zahlen bestehen aus zwei durch Punkt getrennten Ziffernfolgen von denen die erste nicht leer sein darf. Eine reelle Zahl kann durch ein angehängtes E mit anschließender, mit Vorzeichen versehener natürlicher Zahl ergänzt sein. Beispiele: 1987 (ganze Zahl), 3.77E9 (reell, gesprochen als „3.77 mal 10 hoch 9"), 1876 (reell wegen des Punktes). Ganze Zahlen können durch Anhängen von H hexadezimal und durch Anhängen von B oder C oktal interpretiert werden. Zur hexadezimalen Interpretation können auch die Hexziffern ABCDEF mitverwendet werden, allerdings muß die erste Stelle eine Ziffer (0...9) sein. *Bild 1* zeigt die Syntaxdiagramme nach Wirth.

Strings

sind Folgen beliebiger Zeichen, entweder in Anführungsstriche oder in Apostrophe eingeschlossen. Strings mit Apostroph eröffnet und geschlossen, können Anführungen enthalten und umgekehrt. In einem String darf das zugehörige Einschließungszeichen nicht auftauchen.

Operatoren und Begrenzer

sind hier zusammengefaßt in zwei Tabellen, denn sie sind entweder als Sonderzeichen repräsentiert (*Tabelle 1*) oder als bestimmte Wörter aus Großbuchstaben (*Tabelle 2*). Diese letzteren Wörter dürfen nicht als Namen genutzt werden, es sind die reservierten Wörter. Im Vergleich mit Pascal sind einige neue dazugekommen, einige andere fortgelassen (*Tabelle 3*). Das unter den mc-Lesern so umstrittene GOTO fehlt in Modula!

Kommentare

sind Zeichenfolgen, die in „(*" und „*)" eingeschlossen sind. Sie beeinflussen die Bedeutung eines Programmes nicht.
Solche Festlegungen findet man in jedem eine Sprache beschreibenden Text. Die Besonderheit bei der Definition von Modula ist hier, daß Niklaus Wirth betont, daß erst die so gewonnenen Symbole die Elemente

Tabelle 1: Diese Sonderzeichen fungieren als Operatoren oder Begrenzer

+	Addition, Vereinigung von Mengen
-	Subtraktion, Differenz von Mengen
*	Multiplikation, Durchschnitt von Mengen
/	Division, symmetrische Mengendifferenz
:=	Zuweisung
&	logisches Und (AND)
~	logische Verneinung (NOT)
=	gleich
# <>	ungleich
<	kleiner als
>	größer als
<=	kleiner als oder gleich
>=	größer als oder gleich
( )	Klammern
[ ]	Index-Klammern
{ }	Mengen-Klammern
(* *)	Kommentar-Klammern
^	Dereferenzier-Operator
. , : ; |	Satzzeichen

Tabelle 2: Die reservierten Wörter in Modula

AND	ELSIF	LOOP	REPEAT
ARRAY	END	MOD	RETURN
BEGIN	EXIT	MODULE	SET
BY	EXPORT	NOT	THEN
CASE	FOR	OF	TO
CONST	FROM	OR	TYPE
DEFINITION	IF	POINTER	UNTIL
DIV	IMPLEMENTATION	PROCEDURE	VAR
DO	IMPORT	QUALIFIED	WHILE
ELSE	IN	RECORD	WITH

Tabelle 3: Unten die Pascal-Wörter, die nicht in Modula vorkommen, oben umgekehrt

BY	FROM	MODULE
DEFINITION	IMPLEMENTATION	POINTER
ELSIF	IMPORT	QUALIFIED
EXIT	LOOP	RETURN
EXPORT		
DOWNTO	GOTO	PROGRAM
FILE	LABEL	TEXT
FUNCTION	PACKED	

sind, auf der die Syntax von Modula aufbaut. Damit ist die Definition von Modula von den Eigenheiten des Zeichensatzes einer Rechenmaschine frei. Solche Eigenheiten sind mit ein Grund, daß es so viele Basic-Varianten gibt (man denke nur an die Commodore-Zeichensätze bei den Heimcomputern). In der Definition eines Modula-Namens oben ist enthalten, daß Groß- und Kleinbuchstaben unterschieden werden. Alpha ist als Name von alpha verschieden. Außerdem ist die Länge eines Namens nicht begrenzt.

Alles muß vereinbart sein

In diesem Überblick werden die weiteren Sprachkonstrukte von Modula nur kursorisch angesprochen. Die exakten (induktiven) Definitionsschemata in EBNF sind in den Lehrbüchern und den Handbüchern zu den im Handel erhältlichen Compilern zu finden. Hier geht es eher darum, Neugierigen, die gern nur einmal kurz gesehen hätten, wie in etwa Modula funktioniert, eine erste Ahnung zu verschaffen. Anders als in Basic, wo innerhalb eines Programmes beliebig neue Namen genannt werden dürfen, die dann automatisch mit dem entsprechenden Speicherplatz (= Objekt) verbunden sind, darf man in Modula keinen Namen verwenden, der nicht zuvor vereinbart wurde. Dies ist eigentlich der natürlichere Weg gegenüber Basic, denn man muß ja wissen, wovon man spricht. Das ist in Modula besonders wichtig, weil man dort den Bestand an Datentypen und anderen Programmdetails konstruktiv erweitern kann.

Tabelle 4: Die Standardfunktionen

ABS(x)	Absolutwert; Typ des Resultats = Typ des Arguments.
CAP(ch)	wenn ch ein Kleinbuchstabe ist, der entsprechende Großbuchstabe; wenn ch ein Großbuchstabe ist, derselbe Buchstabe.
CHR(x)	das Zeichen mit der Ordnungszahl x. CHR(x) = VAL(CHAR,x)
FLOAT(x)	x vom Typ CARDINAL in einen Wert des Typs REAL konvertiert.
HIGH(a)	obere Indexgrenze von Array a.
MAX(T)	maximaler Wert des Typs T.
MIN(T)	minimaler Wert des Typs T.
ODD(x)	x MOD 2 # 0.
ORD(x)	Ordnungszahl (des Typs CARDINAL) von x. x ist in einer Menge von Werten enthalten, die durch den Typ T von x definiert ist. T ist ein beliebiger Aufzähltyp, CHAR, INTEGER oder CARDINAL.
SIZE(x)	Anzahl von Speicherelementen, die für eine Variable x benötigt werden.
TRUNC(x)	Ganzzahliger Anteil (des Typs CARDINAL) der reellen Zahl x.
VAL(T,x)	der Wert mit Ordnungszahl x und Typ T. T ist irgendein Aufzählungstyp, CHAR, INTEGER oder CARDINAL. VAL(T,ORD(x)) = x, wenn x vom Typ T ist.
DEC(x)	x := x − 1
DEC(x,n)	x := x − n
EXCL(s,i)	s := s − {i}
HALT	beendet die Programmausführung
INC(x)	x := x + 1
INC(x,n)	x := x + n
INCL(s,i)	s := s + {i}

In Basic gibt es halt nur die Standardtypen. Bei der Vereinbarung eines Namens wird der Bezug zwischen Objekt und Namen hergestellt. Objekte können Konstanten, Variablen, Prozeduren, Typen, Module und anderes mehr sein. Nach der Deklaration verweist der Name auf das Objekt. Im Programm GGT aus dem vorigen Teil sind die Namen x, y, u, v mit Objekten vom Typ CARDINAL in der VAR-Deklaration verbunden worden und verweisen danach auf eine solche Zahl. Solche Vereinbarungen können für bestimmte Programmbereiche gelten und in anderen nicht. Das „Wie" bestimmen dabei Sichtbarkeitsregeln, auf die später eingegangen werden muß, wenn es um das modulare Konzept und um Prozeduren geht.

Zum Bestand der Sprache gehören Standardnamen, die von vorneherein als vereinbart gelten. *Tabelle 4* zeigt sie. Sie sind überall gültig (sichtbar). Einige sind aus Pascal bekannt, andere neu.

Die Konstanten-Deklaration

Die Konstanten-Deklaration verbindet einen Namen mit einer Konstanten. Die Konstante selbst ist im einfachsten Fall als Zahl, String oder aufgezählte Menge angegeben, kann aber auch komfortabel durch einen Ausdruck angegeben werden, der einen geeigneten Wert (eine Konstante) liefert. Beispiele nach Wirth:

N = 100
limit = 2*N − 1
all = {0..WordSize-1}

Typ-Deklaration

Nach einer Typ-Deklaration ist der Wertebereich festgelegt, den Variablen dieses Typs annehmen können. Es gibt folgende Typen:

SimpleType, ArrayType, RecordType, SetType, PointerType, ProcedureType,

wobei die einfachen Datentypen aus unstrukturierten Werten bestehen, die nicht selbst wieder aus Werten noch einfacherer Datentypen zusammengesetzt sind. In Modula sind folgende einfachen Datentypen vordefiniert:

BOOLEAN, CHAR, CARDINAL, INTEGER, REAL, (LONGINT), BITSET.

Im Unterschied zu Pascal sind hier CARDINAL, BITSET und LONGINT neu. BITSET ist intern als Bitmuster von Wortbreite dargestellt. Es werden damit alle Teilmengen der Menge des Kardinalzahlabschnittes von 0 bis „Anzahl der Bits in einem Wort" darstellbar. (Eine Kardinalzahl ist genau dann Element, wenn das Bit mit ihrer Nummer gesetzt ist.) Der Wertebereich von CARDINAL sind alle Zahlen, die bei vorzeichenloser Darstellung in einem Maschinenwort zur Verfügung stehen. Bei 16 Bit Wortbreite zum Beispiel 0...65535. Bei 16-Bitsystemen gibt es meist den Typ LONGINT. Er überstreicht den Bereich, der mit Vorzeichen in einem Bit von zwei Wörtern dargestellt werden kann. SympleType enthält noch den Aufzählungstyp, der durch eine Liste von Elementen (die in runden Klammern eingeschlossen sind) dargestellt wird. Die Listenelemente sind Namen für die Ordnungsnummer ihres Auftretens in der Aufzählung. Sie werden als Konstante verwendet. Beispiel: (Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Sonnabend, Sonntag). Zu den einfachen Typen gehören auch Unterbereichstypen, die einen zusammenhängenden Ausschnitt aus den „CARDINALs" oder „INTEGERs" bilden.

ARRAY-Typen ...

... sind in Modula nicht wesentlich anders als in Pascal definiert. Es gibt keine packed arrays.
Beispiel: Vektortyp = ARRAY [0..10] OF REAL;

RECORD-Typen ...

... sind nicht wesentlich anders definiert als in Pascal. Sie können mehrere variante Teile besitzen.
Beispiel: RECORD day: [1..31];
month: [1..12];
year: [0..2000];
END
Es sind variante Records ohne Tagfield-Variable definierbar. Dann wird immer die Variante benutzt, deren Selektor in der Ansprache des Records auftritt.

SET-Typen ...

... sind wie in Pascal definierbar, mit der Einschränkung, daß als Basistyp nur Aufzählungstypen und der Unterbereich 0...N-1 der Zahlen zugelassen sind und meist die Anzahl N der Bits im Maschinenwort die Mächtigkeit begrenzt. BITSET ist vordefiniert.
Beispiel: Wochentage = SET OF (Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag, Sonntag);

POINTER-Typen ...

... sind wie in Pascal definierbar, allerdings durch ein Schlüsselwort POINTER TO anstelle des ↑.

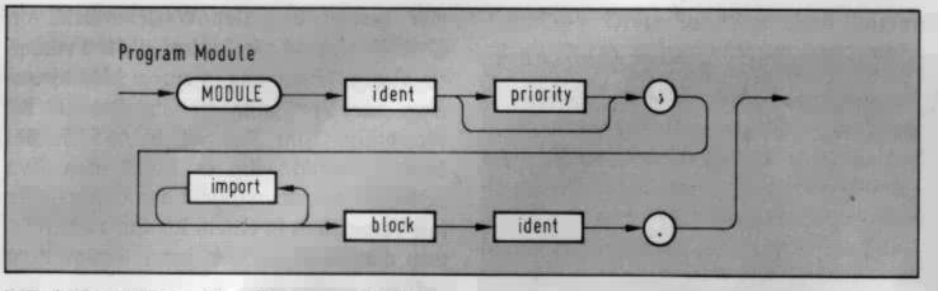


Bild 2. Der Aufbau eines Programm-Moduls

PROCEDURE-Typen ...

...sind völlig neu in Modula. Man kann Variablen vom Typ PROCEDURE vereinbaren, die später eine passende Prozedur zugewiesen bekommt. Darüber später mehr.

Variablen-Deklarationen

Wiederum wie in Pascal kann man durch die Deklaration von Variablen Objekte fester Struktur und festen Datentyps für den Sichtbarkeitsbereich der Vereinbarung über einen Namen ansprechbar machen. Die Vereinbarung einer Variablen enthält dementsprechend den Namen und den Typ der Variablen, der im Falle von Arrays, Records und Sets in der Variablendeklaration näher spezifiziert sein kann.

Die bisher aufgezählten Deklarationsarten werden durch die Schlüsselwörter CONST, TYPE, VAR eingeleitet. Eine durch solch ein Schlüsselwort eingeleitete Definition von Variablen kann eine ganze Liste solcher Deklarationen enthalten. Beispiele:

```
CONST
a = 3;
b = 5;
Toleranz = 1.0E-7;

TYPE
Vektor = ARRAY [0..100] OF REAL;
HundertundeinsREALS = POINTER TO
Vektor;

VAR
KleinerVektor: ARRAY [1..3] OF REAL;
Zeiger: HundertundeinsREALS;
x,y,z,:REAL;
i,j,k,l:CARDINAL;
```

Prozedur-Deklarationen

Auch Prozedurdeklarationen verlaufen zunächst wie in Pascal.
Beispiel:

```
PROCEDURE machwas (VAR x,y:
INTEGER);
VAR z : INTEGER;
BEGIN
z:=x; x:=y; y:=z;
END machwas;
```

Hier ist auch zu sehen, daß wie in Pascal Zuweisungen vorgenommen werden. Wie in Pascal ist die Prozedur mit einem Kopf und einem anschließenden Block aufgebaut. Der Kopf umfaßt das Schlüsselwort PROCEDURE, den Namen der Prozedur, und die formalen Parameter, hier x und y, die beide VAR-Parameter sind. Das bedeutet, daß dieses Unterprogramm beim Aufruf die Adressen der aktuellen Parameter übernimmt und als Referenz benutzt. Das heißt, daß das Unterprogramm direkt die Variablen aus dem übergeordneten Programm manipuliert. Man kann auch eine Wertübergabe des Variableninhaltes an das Unterprogramm erreichen, wenn man VAR fortläßt und nur x,y: INTEGER vereinbart. Dann allerdings sind die neuen Werte von x und y in diesem Beispiel von außen nicht mehr zugänglich, weil ihr Wert von der Prozedur nicht weitergegeben wird. Für Kenner sei hier mitgeteilt, daß Prozeduren auch Arrays als Parameter enthalten können, deren weitere Charakteristika erst zum Übergabezeitpunkt feststehen. Das sind die sogenannten „Open Arrays", die es in Pascal nicht gibt. Der Programmierer ist hier für die Passung verantwortlich.
Der Prozedur-Block besteht aus dem Vereinbarungsteil und dem anschließenden Anweisungsteil, der mit BEGIN eröffnet wird. Das Schlüsselwort END mit angehängtem Prozedurnamen schließt die gesamte Prozedur ab.

Ein Modul

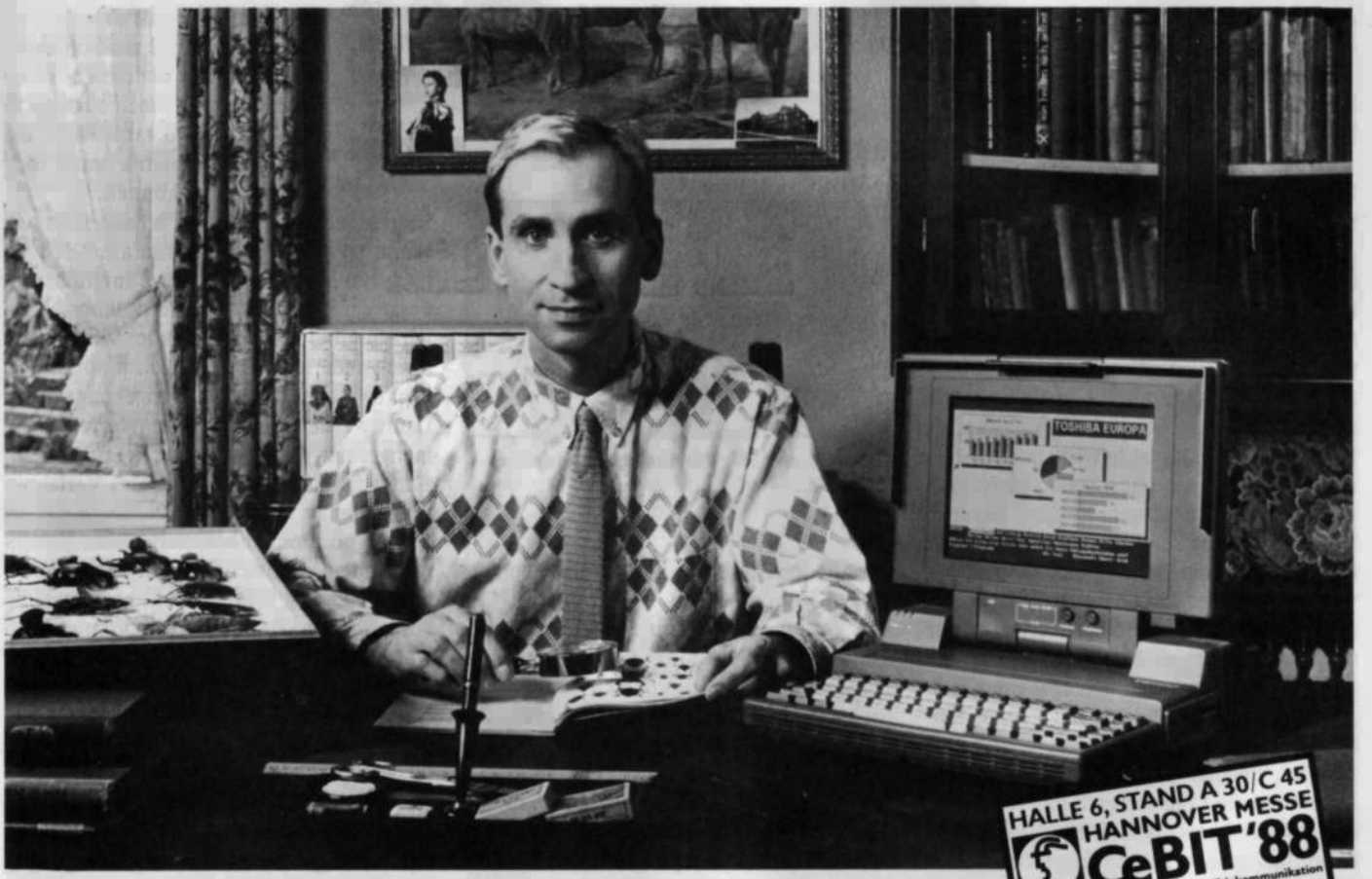
Bisher wurden nur Dinge über Modula gesagt, die Ihr Vertrauen darin stärken sollten, daß Modula auch wie Pascal ist. Und daß das erste Programmieren in Modula gerade für Pascalfreunde nur solange schwierig ist, solange man sich in den kleinen, feinen Unterschieden verheddert, die es in manchen Formalia gibt. Das eigentliche Programmieren in Modula ist dann aber von weiterreichenden Mechanismen bestimmt als in Pascal. Ein Programm in Modula besteht in seiner obersten Hierarchiestufe aus einem Programm-Modul, der fast wie ein Pascalprogramm aussieht. Anstelle des Schlüsselwortes PROGRAM steht allerdings das Wort MODULE. Ein Beispiel:

```
MODULE mc;
FROM Terminal IMPORT WriteString,
WriteLn;

CONST mcUntertitel = „die Mikrocomputer-Zeitschrift";

BEGIN
WriteString(mcUntertitel);
WriteLn;
END mc.
```

Es kommt also nach dem eröffnenden Schlüsselwort MODULE der Name des „Programmmoduls". Dann folgt eine Zeile, die eine Importliste enthält. In der genannten Form enthält sie Namen von Prozeduren und Objekten, die in bestimmten anderen Modulen (das sind dann keine Programm-Module) definiert sind (hier ist es der Modul Terminal) und von diesen exportiert werden (was nichts anderes heißt, als daß sie zur allgemeinen Benutzung zur Verfügung gestellt werden). Man kann auch komplette Module importieren. Das wird alles später noch vorgeführt.
Dann folgt im Block des Moduls eine Konstanten-Definition in der ein String mit dem Namen mcUntertitel verbunden wird. Durch das weitere Programm wird dieser String dann ausgedruckt. Das Bemerkenswerte am Konzept von Modula ist, daß Mechanismen existieren, die es erlauben, Module gewissermaßen auf Vorrat zu fertigen, diese in eine Bibliothek aufzunehmen und damit in das Modulasystem so zu integrieren, daß diese vorgefertigten Module oder Objekte daraus von anderen Modulen benutzt werden können. Manche Standard-Prozeduren aus Pascal sind in Modula nicht mehr in den Compiler integriert, sondern in Modulen untergebracht, von wo sie nach Bedarf importiert werden müssen. Dies gilt insbesondere für die I/O-Prozeduren, die ja weitgehend von der Implementation abhängen. In Modula können sie leicht geändert und an die unterliegende Maschine angepaßt werden. Im Beispiel oben WriteString und WriteLn, deren Name schon für sich selbst spricht. Das wird im nächsten Teil der Serie genauer betrachtet werden. Das Ziel in diesem Teil war nur, Ihnen das Gefühl zu geben, daß sich Modula beim ersten Programmieren wie Pascal anfühlt. Niklaus Wirth hätte aber Modula nicht erfunden, wenn das schon alles gewesen wäre. Über das weitere wird in der nächsten Ausgabe berichtet. *Wird fortgesetzt*

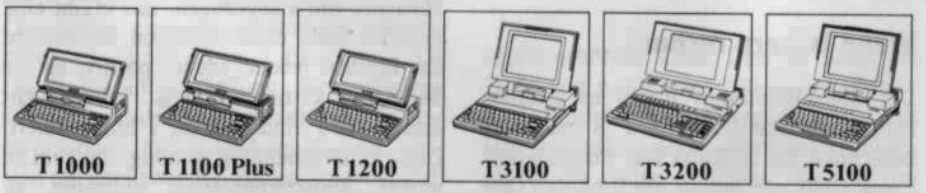
Wissen ist Macht.

Immer mehr Speicherkapazität und eine immer schnellere Zugriffszeit zu besitzen, jederzeit darüber verfügen zu können und kompatibel zum Weltstandard zu sein, das sind die Vorteile tragbarer Computer. Ihnen gehört die Zukunft.
TOSHIBA will diesen technologischen Vorsprung mit seinen Portable Computern allen schon heute zur Verfügung stellen.



80386/40MB, das Herz des T 5100 von TOSHIBA.

BERATUNGSGUTSCHEIN



Hier sehen Sie alle TOSHIBA Portable Computer auf einen Blick. Um genau zu wissen, welcher der Richtige ist, sollten Sie sich beraten lassen.
Wenn Sie Informationsmaterial über die Portables von TOSHIBA haben wollen, schicken Sie den ausgefüllten Beratungsgutschein bitte an TOSHIBA.
Oder wenden Sie sich direkt an einen autorisierten TOSHIBA Fachhändler, dort können Sie die Portables auch im Original sehen.

Firma/Branche

Name

Straße

PLZ Ort

Telefon

MC 388/51

In Touch with Tomorrow

TOSHIBA Europa (I.E.) GmbH, Bereich IPS Deutschland, Hammer Landstraße 115, 4040 Neuss.

Eckart Winkler

Dreifach genäht

Der RSA-Algorithmus verspricht optimale Sicherheit bei der Verschlüsselung

Die Codierung von Informationen ist so alt wie die Schrift; die Verfahren wurden jedoch immer wieder aufgedeckt. Der RSA-Algorithmus arbeitet mit drei Schlüsseln, zwei öffentlichen und einem geheimen, der nicht aus den beiden anderen rekonstruiert werden kann.

Die Mathematiker Ronald Rivest, Adi Shamir und Leonard Adleman stellten 1978 ein ganz neues Verfahren zur Codierung und Decodierung von Informationen vor. Das nach den Anfangsbuchstaben ihrer Nachnamen „RSA-Algorithmus" genannte Verfahren arbeitet mit drei Schlüsseln, davon sind zwei öffentlich. Das Besondere daran ist: Jeder, der die öffentlichen Schlüssel kennt, ist in der Lage, Daten zu codieren. Aber nur wer den dritten Schlüssel besitzt, kann die verschlüsselten Daten entschlüsseln. Obwohl dieser dritte Schlüssel mit den beiden anderen in gewisser Beziehung steht, ist es nicht möglich, ihn zu rekonstruieren: das ist bewiesen.

Vor dem Verfahren selbst sollen einige mathematische Grundlagen geklärt werden; am Schluß steht seine Programmierung in Pascal. Allerdings muß gleich angemerkt werden, daß der Original-Algorithmus mit sehr großen Zahlen arbeitet, etwa in der Größenordnung von 100 Stellen. Wir werden uns jedoch mit den vorgegebenen Integer-Zahlen begnügen. Das hat keinen Einfluß auf die Funktionsfähigkeit des Programms, jedoch werden Außenstehende dadurch in die Lage versetzt, den fehlenden Schlüssel zu berechnen. Dies ist bei außerordentlich großen Zahlen ausgeschlossen.

Das Verfahren beruht auf dem Rechnen mit Resten. Bei einer Division innerhalb der ganzen Zahlen entsteht ja dann ein Rest, wenn der Divisor nicht Teiler des Dividenden ist. Soll etwa 11 durch 3 geteilt werden, so ist das Resultat 3, jedoch bleibt ein Rest von 2. Um solche Reste zu berechnen, existiert in Pascal der Operator mod. Auf das Beispiel angewandt: 11 mod 3 ergibt 2. Grundlegend ist nun die Tatsache, daß man mit derartigen Resten auch rechnen kann. Angenommen, es soll der Rest des Produktes zweier Zahlen bezüglich der Division durch 7 berechnet werden. Es gibt zwei Wege; am nächsten liegt, zuerst das Produkt zu bilden und dann den erforderlichen Rest zu berechnen. Es geht jedoch auch anders: Man ermittelt zunächst die Reste der beiden Zahlen und multipliziert diese miteinander. Eventuell muß vom Ergebnis noch einmal der Rest gebildet werden.

```
Zu berechnen ist 51*29.

  a       b     ungerade? notieren
 51      29        j          51
102      14        n
204       7        j         204
408       3        j         408
816       1        j         816
                                ----
                  Summe:      1479
```

Bild 1. Beispiel für die russische Bauernmultiplikation

Dann jedoch sind die Resultate bei beiden Wegen identisch.

Der Vorteil der zweiten Methode ist, daß die bei der Rechnung anfallenden Zwischenergebnisse wesentlich kleiner sind als bei der ersten Methode. Ein Beispiel: Zu berechnen sei 129*57 mod 7. Hier kommt ein Pascal-Compiler auf einem PC bereits in Schwierigkeiten, denn 129*57=123453 übersteigt den Rechenbereich, der meist bei 32767 endet. Besser geht es, wenn erst 129 mod 7 und 957 mod 7 berechnet werden. Das ergibt 3 und 5 und das Produkt 15, durch Restbildung erhält man das Endergebnis 1.

Wenn Bauern rechnen ...

Der RSA-Algorithmus macht es erforderlich, Ausdrücke der Form a↑s mod n zu berechnen (↑ bedeutet hier Potenzieren). Pascal stellt, wie bereits erwähnt, den Operator mod zur Verfügung, jedoch muß der linke Operand dabei zunächst immer einzeln ausgerechnet werden. Dies würde in den meisten Fällen zu einem Überlauf führen. Deshalb soll eine direkte Berechnung von a↑s für den Algorithmus vermieden werden. Zunächst soll allerdings a*b mod n näher betrachtet werden.

Zur Multiplikation wird oft ein Algorithmus verwendet, der unter dem Namen „Russische Bauernmultiplikation" bekannt ist. Den russischen Bauern früherer Zeiten sagte man nämlich nach, daß sie außer addieren nur mit 2 multiplizieren und durch 2 dividieren konnten. Mehr ist jedoch auch nicht notwendig, um tatsächlich alle beliebigen Zahlen a und b multiplizieren zu können.

Hierzu wird geprüft, ob b ungerade ist. Wenn ja, wird der Wert von a notiert. Im nächsten Schritt wird a mit 2 multipliziert, b durch 2 geteilt. Ein auftretender Rest wird hierbei ignoriert. Wieder wird, wenn b ungerade ist, der Wert von a notiert. So geht es weiter, bis b bei 0 angekommen ist. Dann werden alle notierten Werte zusammengezählt, und man hat das Ergebnis. In *Bild 1* ist ein Beispiel für dieses Verfahren angegeben.

Der Algorithmus modifiziert dieses Verfahren, indem er gleich nur mit Resten rechnet. Im Listing (*Bild 5*) steht als erstes die Funktion abmodn. Die Werte von a und b werden nach x und y übertragen. Es ist sinnvoll, die kleinere Zahl in y zu speichern, da die Größe von y die Laufzeit des Algorithmus bestimmt. Deshalb die Fallunterscheidung am Anfang. In der Variablen s werden die notierten Werte gleich aufsummiert. Hier wird also nur etwas addiert, wenn y ungerade ist, wie die Bedingung (y mod 2)=1 ausdrückt.

Zu beachten ist die Unterscheidung bei der Ermittlung des neuen Werts von s. Sollte x+s den Wert von n übersteigen, wird n zunächst von s abgezogen. Das ergibt eine negative Zahl, durch Addition von x wird das Resultat jedoch wieder positiv, da die Bedingung x>n–s erfüllt ist. Ist sie nicht erfüllt, wird einfach x zu s hinzuaddiert. Diese Unterscheidung ist nötig, um kein zu großes Zwischenergebnis auftreten zu lassen.

Eine ähnliche Fallunterscheidung muß bei der Modifikation von x vorgenommen werden. Auch hier wird zunächst n subtrahiert, falls x+x zu groß ist. Lediglich bei der Änderung von y ist nichts zu beachten. Die ganzzahlige Division mittels div kann ohne Fallunterscheidung erfolgen.

Die Funktion, die für den Algorithmus gebraucht wird, heißt ahsmodn. Sie berech-

net a↑s mod n. Da sie die Funktion abmodn benutzt, wird kein Überlauf verursacht. Auch hier wird x jeweils mit 2 multipliziert, y durch 2 dividiert. r wird jeweils nur dann mit x multipliziert, wenn y ungerade ist. Man kann sich leicht von der Korrektheit dieser Funktion überzeugen, wenn man an die Binärdarstellung von y denkt: z. B. ist a↑13=a↑1∗a↑4∗a↑8. Die hier vorkommenden Potenzen sind genau diejenigen, die von der Funktion ahsmodn berechnet werden.

Zunächst muß nun klar sein, was der Algorithmus leistet: Er ist in der Lage, eine Zahl bis zu einer bestimmten Größe auf eine andere Zahl abzubilden und aus dieser die Ursprungszahl wieder eindeutig zu rekonstruieren. Mit anderen Worten werden also Zahlen codiert und wieder decodiert. Meist ist man jedoch an der Verschlüsselung von Texten interessiert. Hier wird man beispielsweise auf den ASCII-Code der vorkommenden Zeichen zurückgreifen. Sinnvollerweise faßt man immer die ASCII-Codes mehrerer Zeichen zusammen, sonst könnte ein Abzählen der Häufigkeit der im Code vorkommenden Zahlen einen Anhaltspunkt für ihre Bedeutung liefern.

Im Programm muß wieder ein Zugeständnis an den zu kleinen Rechenbereich gemacht werden: Es wird nicht der ASCII-Code verwendet sondern einer, der die Zahl der verfügbaren Zeichen stark einschränkt. Bei diesem Code sind nur die Buchstaben und die Zeichen „.: sowie das Leerzeichen vorhanden. Damit das Progamm bei Auftreten eines unzulässigen Zeichens trotzdem arbeitet, werden die Kleinbuchstaben in Großbuchstaben umgewandelt, aus allen anderen nicht vorgesehenen Zeichen wird ein Leerzeichen. Die Codes liefert die Funktion cd. Die Umkehrung, die für die Decodierung benötigt wird, wird durch die Funktion dc vollzogen.

Jeweils zwei Codes c1 und c2 werden zu einer Zahl zusammengefaßt, und zwar durch c=c1∗32+c2. Aus c können c1 und c2 wieder eindeutig gewonnen werden durch c1=c div 32 und c2=c mod 32. Durch die Einschränkung der zulässigen Zeichen und das Zusammenfassen von nur zwei Codes (üblich sind mindestens drei) sind die entstehenden Zahlen auf jeden Fall kleiner als 1000, so daß keine Probleme entstehen.

Wie funktioniert das Verfahren jetzt? Angenommen, wir seien diejenigen, die verschlüsselte Botschaften empfangen und decodieren wollen. Dann suchen wir uns zwei Primzahlen p und q aus. Diese sollten beide etwa 50 Stellen besitzen. Die Zahl n=p∗q geben wir öffentlich bekannt. Wesentlich ist, daß niemand die beiden Primzahlen ermitteln kann. Dies ist durch die Größe von n mit etwa 100 Stellen absolut gewährleistet. Eine Zerlegung auch mit dem schnellsten derzeit verfügbaren Rechner würde einige Millionen Jahre in Anspruch nehmen.

Weiter berechnen wir f=(p−1)∗(q−1). f bleibt geheim. Schließlich suchen wir uns noch die Zahlen s und t aus. s muß kleiner als f sein und darf mit f keinen Teiler gemeinsam haben. An t wird die Anforderung s∗t mod f = 1 gestellt. Zum Verschlüsseln wird s benötigt, s wird also bekanntgegeben. t bleibt jedoch geheim, nur mit t ist die Decodierung möglich.

Unter diesen Voraussetzungen geschieht die Verschlüsselung einer Zahl x einfach durch y=x↑s mod n. Die Entschlüsselung wird genauso einfach mit x=y↑t mod n durchgeführt. Die Korrektheit dieses Verfahrens soll hier nicht bewiesen werden. Wer daran sowie an weiteren Einzelheiten interessiert ist, kann in der Original-Veröffentlichung [1] nachlesen. In *Bild 2* ist das Wesentliche des RSA-Algorithmus noch einmal übersichtlich zusammengefaßt.

Wie bereits erwähnt, können im Pascal-Programm keine 100stelligen Zahlen auftauchen. Immerhin wird aber durch die geschickte Implementation von abmodn

```
Voraussetzungen:

Gegeben ist eine Folge x(1)...x(m) von Zahlen.  Diese wird in eine
Folge y(1)...y(m) von Zahlen verschlüsselt und entschlüsselt.

Bezeichnungen:

p,q   beliebige Primzahlen                    (geheim)
n     = p*q                                   (öffentlich)
f     = (p-1)*(q-1)                           (geheim)
s     beliebig mit s<f und ggT(f,s)=1         (öffentlich)
t     beliebig mit s*t mod f = 1              (geheim)

Verschlüsselung:

Berechnung von    y(i) = x(i)^s mod n    für i=1...m .

Entschlüsselung:

Berechnung von    x(i) = y(i)^t mod n    für i=1...m .
```

Bild 2. Der RSA-Algorithmus

```
MIT DEM ALGORITHMUS VON RIVEST, SHAMIR UND ADLEMAN KANN
JEDER, DER S UND N KENNT, EINEN TEXT VERSCHLUESSELN.
ABER NUR WER T KENNT, KANN IHN WIEDER ENTSCHLUESSELN.

Codierter Text:

 16910 3653 11014 15518 7894 13584 27204 2492 16362 21048
 17179 21502 27204 12170 5024 10157 30603 112 24186 22019
 16937 8968 24318 7866 17473 16299 17473 21502 0
 28585 11014 1859 24627 23843 2863 22019 16937 23047 16299
 17 13342 10157 28272 16817 21502 15717 22768 26416 23843
 10816 28371 11353 3767 1398 21975 0
 15509 23843 23047 3078 26806 23843 3416 16299 17 13342
 10157 22176 2346 8662 10004 26806 2062 11014 650 17
 14789 21091 24244 20545 12261 30845 4066 0
```

Bild 3. Ein Beispiellauf für das Programm

```
Wähle p=107 und q=293.

Dann ist n=31351 und f=30952.

Wähle nun a=5. Dann ist a*f+1=154761

Primfaktorzerlegungen: f =  2 *  2 *  2 * 53 * 73
                       a*f+1 =  3 * 79 * 653

Wähle z.B. s=3*79=237 und t=653
```

Bild 4. Bestimmung der Schlüssel

der vorhandene Rechenbereich voll ausschöpft. n wird also meist in der Gegend von 30000 liegen.
Ausgeführt werden Ver- und Entschlüsselung von den Prozeduren rsa_code und rsa_decode. Sie lesen die nötigen Daten jeweils aus einem File, bearbeiten sie und schreiben das Resultat in ein anderes File. Die Bearbeitung erfolgt dabei immer zeilenweise. Die definierten Datenstrukturen sind genau darauf ausgerichtet. So kann eine Textzeile in einer Variablen vom Typ linetype aufgenommen werden, für das codierte Äquivalent ist clinetype vorgesehen. Dies ist ein Feld mit halb so vielen Plätzen wie linetype. Mehr ist ja nicht erforderlich, denn es werden jeweils zwei Zeichencodes zu einer Zahl zusammengefaßt. Die 81. bzw 41. Stelle im Feld wird benutzt, um auch bei vollem Feld das Ende leicht erkennen zu können.

Das Programm im einzelnen

Zum Schreiben und Lesen einer Textzeile sind writeline und readline vorgesehen. Beim Lesen wird das Zeilenende mittels der Funktion eof erkannt. readline speichert zur Kennung dann chr(0) auf die nächste Stelle. Erwähnt werden soll noch die Art der Speicherung des Codes im File. Der Code besteht ja aus einer Folge von bis zu fünfstelligen Zahlen. Diese werden in Zehnergruppen in eine Zeile des Files geschrieben, um Übersichtlichkeit zu wahren. Somit reicht eine Zeile zur Aufnahme von 20 Zeichen der Textzeile. Das Ende wird durch eine zusätzliche 0 angezeigt. Aufgrund der Struktur des RSA-Algorithmus kann eine 0 nicht als Code auftreten, so daß das Zeilenende eindeutig markiert wird. Obwohl es sich bei dem Code nur um Zahlen handelt, benutzen wir hier trotzdem auch ein Textfile, damit wir uns das Resultat z.B. mit einem Editor ansehen können. Durch die Modularisierung sind die Codier- und Decodier-Prozeduren nun recht leicht zu verstehen. Beide lesen jeweils das File source und schreiben in das File dest. Die Abbruchbedingung ist in beiden Fällen eof (source). Es wird immer die Zeile sl komplett gelesen und zu dl verarbeitet. Das Ver- und Entschlüsseln geschieht wie besprochen. Am Ende eines jeden Schleifendurchlaufs wird dl in das File dest geschrieben.
Das Hauptprogramm erwartet nun ein Textfile orig.txt. Es wird codiert, das Ergebnis in coded.txt geschrieben. Dann wird dieses durch rsa_decode wieder decodiert, der Text wird in das File decod.txt übertragen. Die Vereinbarungen zum Öffnen der Files müssen evtl. an den verwendeten Compiler angepaßt werden. Hier wurde eine Form

```
program test;
type linetype=array[1..81] of char;
     clinetype=array[1..41] of integer;
var fil1,fil2,fil3:text;
    n,s,t:integer;

function abmodn(a,b,n:integer):integer;
  (* berechnet a*b mdd n ohne Ueberlauf *)
  var x,y,s:integer;
  begin
    if a>b then begin
      x:=a; y:=b end
    else begin
      x:=b; y:=a end;
    s:=0;
    while y>0 do begin
      if (y mod 2)=1 then
        if x>n-s then
          s:=(s-n)+x
        else
          s:=s+x;
      if x>(n div 2) then
        x:=(x-n)+x
      else
        x:=x+x;
      y:=y div 2
    end; (* while *)
    abmodn:=s
  end;

function ahsmodn(a,s,n:integer):integer;
  (* berechnet a^s mod n ohne Ueberlauf *)
  var x,y,r:integer;
  begin
    x:=a; y:=s; r:=1;
    while y>0 do begin
      if (y mod 2)=1 then r:=abmodn(r,x,n);
      x:=abmodn(x,x,n);
      y:=y div 2
    end; (* while *)
    ahsmodn:=r
  end;

function cd(c:char):integer;
  (* codiert ein Zeichen *)
  begin
    if c in ['A'..'Z'] then cd:=ord(c)-64;
    if c in ['a'..'z'] then cd:=ord(c)-96;
    if c=',' then cd:=27;
    if c='.' then cd:=28;
    if c=':' then cd:=29;
    if not (c in ['A'..'Z','a'..'z',',','.',':']) then cd:=30
  end;

function dc(n:integer):char;
  (* decodiert ein Zeichen *)
  begin
    if n in [1..26] then dc:=chr(n+64);
    if n=27 then dc:=',';
    if n=28 then dc:='.';
    if n=29 then dc:=':';
    if n=30 then dc:=' '
  end;

procedure writeline(var f:text;var l:linetype);
  (* schreibt die Zeile f auf das File f mit LF *)
  var i:integer;
  begin
    i:=1;
    while l[i]<>chr(0) do begin
      write(f,l[i]); i:=i+1
    end;
    writeln(f)
  end;

procedure readline(var f:text;var l:linetype);
  (* liest die Zeile l vom File f mit LF *)
  var i:integer;
      c:char;
```

Bild 5. Der RSA-Algorithmus, programmiert in Pascal

```
  begin
    i:=1;
    while not eoln(f) do begin
      read(f,c); l[i]:=c; i:=i+1
    end;
    readln(f); l[i]:=chr(0)
  end;

procedure writecline(var f:text;var c:clinetype);
  (* schreibt eine codierte Zeile auf das File f *)
  var i:integer;
  begin
    i:=1;
    while c[i]<>0 do begin
      if (i<>1) and ((i mod 10)=1) then writeln(f);
      write(f,' ',c[i]); i:=i+1
    end;
    writeln(f,' 0');
  end;

procedure readcline(var f:text;var c:clinetype);
  (* liest eine codierte Zeile vom File f *)
  var i:integer;
  begin
    i:=1;
    repeat
      while not eoln(f) do begin
        read(f,c[i]); i:=i+1
      end;
      readln(f)
    until c[i-1]=0
  end;

procedure rsa_code(var source,dest:text;s,n:integer);
  (* liest das Textfile source, codiert den Text *)
  (* und schreibt das Resultat in das File dest  *)
  var sl:linetype;
      dl:clinetype;
      i,j,m:integer;
  begin
    while not eof(source) do begin
      readline(source,sl); i:=1; j:=1;
      while sl[i]<>chr(0) do begin
        m:=cd(sl[i])*32+cd(sl[i+1]);
        dl[j]:=ahsmodn(m,s,n);
        j:=j+1; i:=i+2;
        if sl[i-1]=chr(0) then i:=i-1
      end;
      dl[j]:=0; writecline(dest,dl)
    end
  end;

procedure rsa_decode(var source,dest:text;t,n:integer);
  (* liest das File source, decodiert die Zahlen *)
  (* und schreibt das Resultat in das File dest  *)
  var sl:clinetype;
      dl:linetype;
      i,j,m:integer;
  begin
    while not eof(source) do begin
      readcline(source,sl); i:=1; j:=1;
      while sl[i]<>0 do begin
        m:=ahsmodn(sl[i],t,n);
        dl[j]:=dc(m div 32); dl[j+1]:=dc(m mod 32);
        i:=i+1; j:=j+2
      end;
      dl[j]:=chr(0); writeline(dest,dl)
    end
  end;

begin
  n:=31243; s:=79; t:=391;
  writeln('*** Codieren ***');
  reset(fil1,'orig.txt');
  rewrite(fil2,'coded.txt');
  rsa_code(fil1,fil2,s,n);
  writeln('*** Decodieren ***');
  reset(fil2,'coded.txt');
  rewrite(fil3,'decod.txt');
  rsa_decode(fil2,fil3,t,n);
  writeln('*** fertig ***')
end.
```

gewählt, die mit einer Anweisung auskommt. Oft wird eine zusätzliche assign-Anweisung erforderlich. Die Resultate eines Beispiellaufs sind in *Bild 3* angegeben. Dies kann zur Kontrolle einer eigenen Implementation dienen, da die Codes bei Benutzung derselben Schlüssel natürlich identisch sein müssen.

Zum Schluß noch einiges zu den drei Schlüsseln n, s und t. Zunächst muß man wissen, wie man zu solchen Zahlen gelangt, die ja die in Bild 2 angegebenen Bedingungen erfüllen müssen. Zwar ist es leicht, sich zwei Primzahlen auszusuchen und daraus n zu berechnen. Auch ein s läßt sich noch leicht finden. Jedoch stellt sich meist heraus, daß zu diesem s kein entsprechendes t existiert. Folgende Methode erweist sich als erfolgreich.

Drei Schlüssel im Bunde

Wähle zwei Primzahlen p und q (p*q darf natürlich nicht den Rechenbereich überschreiten). f kann leicht berechnet werden. Wähle nun eine beliebige Zahl a. Ermittele die Primfaktorzerlegungen von f und (a*f+1). Verteile die Primfaktoren von (a*f+1) gleichmäßig auf s und t, kein Faktor darf übrigbleiben. Für s dürfen natürlich nur diejenigen Faktoren verwendet werden, die in der Zerlegung von f nicht auftreten. *Bild 4* zeigt ein Beispiel für die Durchführung dieses Verfahrens; es muß ein Taschenrechner benutzt werden, weil (a*f+1) fast immer zu groß ist (außer wenn a=1 gewählt wird).

Schließlich ist noch die Ermittlung von t bei Kenntnis der Primzahlen p und q interessant. Mit p und q ist ja f bekannt. Genau wie eben kann dann eine Zahl a gesucht werden. Geeignet ist ein a, für das s ein Teiler von (a*f+1) ist. Dann nämlich gilt t=(a*f+1)/s.

Bei den im Pascal-Programm verwendeten n ist es natürlich sehr einfach, p und q zu ermitteln und so auch t herauszubekommen. In Modula-2 sieht das schon anders aus. In dieser Sprache stehen meist Moduln für das Rechnen mit ganzen Zahlen beliebiger Größe zur Verfügung. Es ist relativ einfach, ein Pascal-Programm in Modula-2 zu formulieren. Der einzige Nachteil wäre dann die immens hohe Rechenzeit, die die des Pascal-Programms um Größenordnungen übertreffen würde.

Literatur

[1] *Rivest, Shamir, Adleman:* A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. In der Zeitschrift Communications of the ACM. Heft 21/1978, S.120–126

Hagen Völzke

Digitale Bilderzeugung und -verarbeitung

Teil 4: Eine Grafikkarte für den mc-68000-Computer

Im letzten Heft wurden verschiedene Grafikprozessoren mit unterschiedlicher Leistungsfähigkeit vorgestellt. Diesmal nun geht es um eine Applikation: Am Beispiel einer Grafikkarte für den mc-68000-Computer wird der Einsatz des GDP 82786 von Intel mit der Motorola-CPU 68000 demonstriert.

Intels Grafikprozessor wurde ursprünglich zum Einsatz mit den Prozessoren 8086, -186, -286 und -386 konzipiert, um moderne Grafikstandards wie GKS, NAPLPS und VDI (Virtual Device Interface) zu unterstützen. Er kann sowohl mit synchronen (wie den oben erwähnten) als auch mit asynchronen Mikroprozessoren (wie z.B. dem MC 68000) zusammenarbeiten. Daher läßt sich der GDP 82786 relativ einfach im mc-68000-Computer einsetzen. Ein Blick auf den Schaltplan (*Bild 1*) verrät schon, daß der GDP 82786 zu recht in die Klasse der 'VLSI'-Chips fällt: Die gesamte Steuerlogik für dynamische RAMs, für die Speicherarbitrierung (arbiter = Schiedsrichter; Zuteilung der Zugriffsberechtigung an verschiedene Prozessoren) und für die Pixelaufbereitung befindet sich bereits auf dem Chip. Braucht man nur einen Monitor mit TTL-Eingang, so zeigt Bild 1 bereits die komplette Schaltung, da der GDP 82786 einen Monitor direkt ansteuern kann.

Je nach angestrebter Auflösung, Bildfrequenz, Farbanzahl und Speicherkapazität sind sehr unterschiedliche Konfigurationen möglich. *Tabelle 1* zeigt einige technische Daten, die sich mit dieser Schaltung realisieren lassen. Noch höhere Auflösungen, das Parallelschalten mehrerer Grafikprozessoren oder Auswerten bestimmter Fensterattribute werden hier nicht beschrieben und können den Datenblättern der Bausteine entnommen werden (→ Literaturangaben).

Die Schaltung

Besprechen wir nun einige Details der Schaltung: Die Prozessoren von Intel verfügen über die Adreßleitungen A0...Axx (je nach Typ). Die CPU 68000 hingegen zählt erst ab Adreßleitung A1. Dafür liefert sie aber die Signale $\overline{\text{LDS}}$ und $\overline{\text{UDS}}$, die CPU-intern aus der Adreßleitung A0 und der Information, ob ein Byte- oder ein Wortzugriff stattfinden soll, abgeleitet werden.

Die Intel-Prozessoren erledigen dies über die Signale A0 und $\overline{\text{BHE}}$ (Byte High Enable). Praktischerweise entspricht das Signal $\overline{\text{LDS}}$ genau dem A0-Signal einer Intel-CPU und das Signal $\overline{\text{UDS}}$ dem $\overline{\text{BHE}}$-Signal. Wie vielleicht bekannt ist, speichern Intel-CPUs die Bytes von 16-Bit-Worten genau in umgekehrter Reihenfolge gegenüber dem Prozessor 68000. Dies könnte nun bei Wortzugriffen auf den Grafikprozessor zu Schwierigkeiten führen. Tatsächlich gibt es aber keine Probleme, da bei Intel nicht nur die Werte in der anderen Reihenfolge im Speicher liegen, sondern gleichzeitig A0 mit inverser Bedeutung zur Adreßleitung A0 der CPU 68000 betrieben wird – umdenken bei Byte- und Wort-Zugriffen auf den Grafikprozessor ist also unnötig. Lediglich Langworte (die der GDP 82786 hin und wieder zum Adressieren seines Speichers benötigt) müssen vor der Verarbeitung mit dem 'SWAP'-Befehl durch den Prozessor 68000 korrigiert werden.

Der Reset

Ein Monoflop (74 LS 122) erzeugt das im Datenblatt geforderte RESET-Signal für den GDP 82786. Beim Einschalten des Compu-

Tabelle 1: Technische Daten

Auflösung bei maximaler Farbzahl

- 640 x 480 Punkte, 60 Hz Bildfrequenz, 256 Farben pro Punkt, auswählbar aus einer Palette von 256000 Farben bei 25 MHz Punktfrequenz, Darstellung non-interlaced oder
- 768 x 512 Punkte, 60 Hz Bildfrequenz, bei 35 MHz Punktfrequenz, Darstellung non-interlaced

Maximale Auflösung

1024 x 800 Punkte, 60 Hz Bildfrequenz, 16 Farben pro Punkt, 70 MHz Punktfrequenz Darstellung non-interlaced

Maximaler Speicher

4 Megabyte bei Verwendung von 1 MBit Chips oder 1 Megabyte bei Verwendung von 256 KBit Chips

Die maximale Punktfrequenz des Graphikprozessors beträgt 35 MHz

Plotgeschwindigkeit

Linien, Polygone:
2,5 Mio Pixel/Sekunde
Kreise, Bogenstücke:
2,0 Mio Pixel/Sekunde
Füllen:
30 Mio Bit/Sekunde
Bit-Block-Transfer:
24 Mio Bit/Sekunde
Zeichendarstellung:
25000 Zeichen/Sekunde

Zeichensätze

Beliebige Zahl von Zeichensätzen, Proportionalschrift, verschiedene Zeichensatzgrößen usw.

Fenster

Hardwaremäßig, Anzahl im Grunde nur durch Gesamtspeicher beschränkt. Scrolling und Panning punktweise möglich, keine externe Hardware nötig.

Cursor

16 x 16 Pattern oder Fadenkreuz hardwaremäßig (Overlay)
Unterstützung für GKS, NAPLPS und VDI Standards.
Bildformate, Ablenkfrequenzen, Bittiefe usw. frei programmierbar.

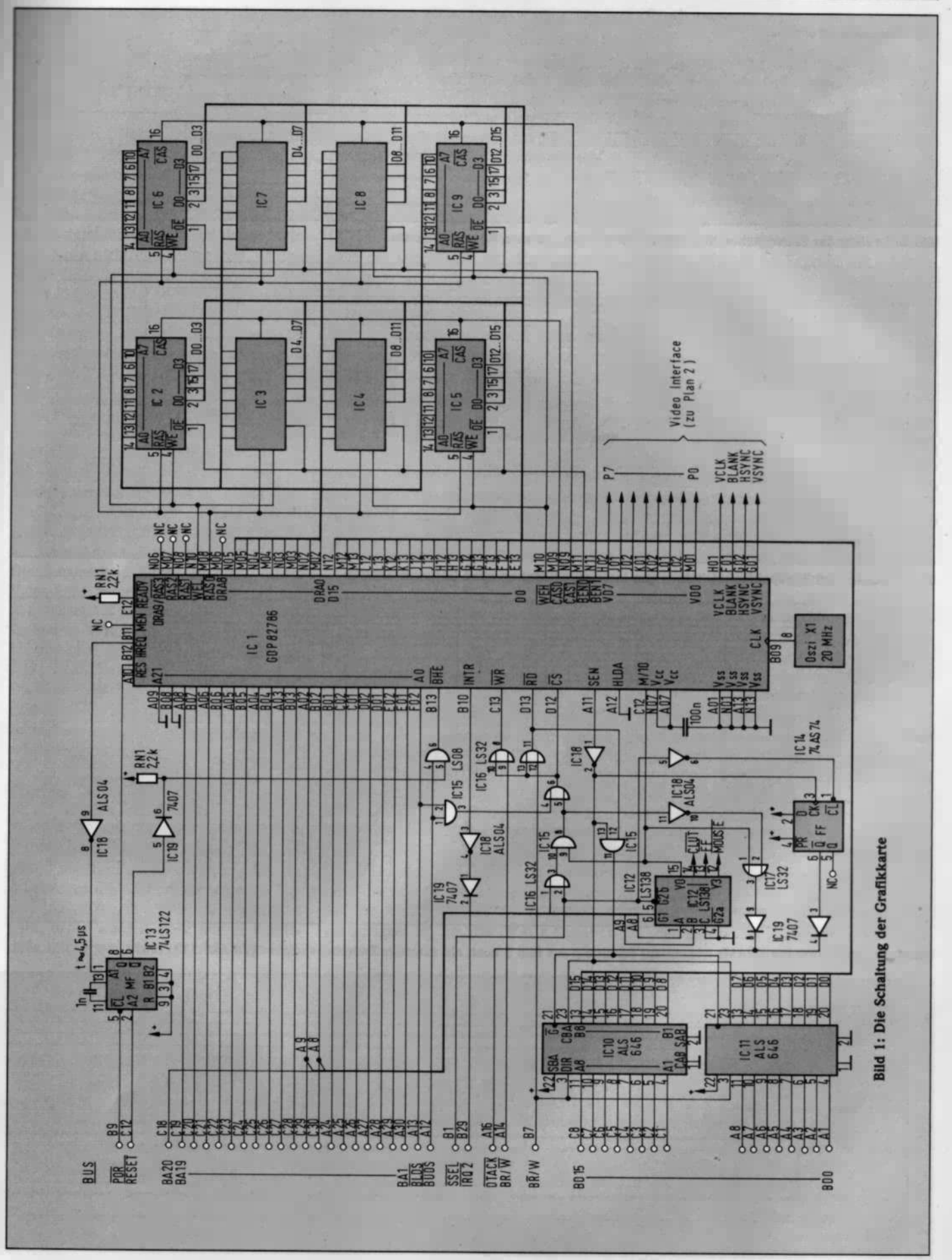


Bild 1: Die Schaltung der Grafikkarte

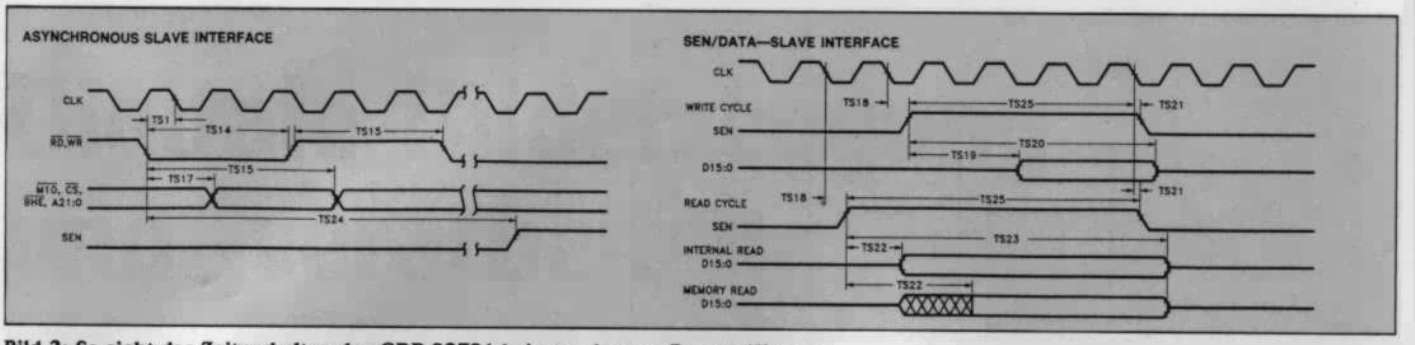


Bild 2: So sieht das Zeitverhalten des GDP 82786 bei asynchronen Buszugriffen aus

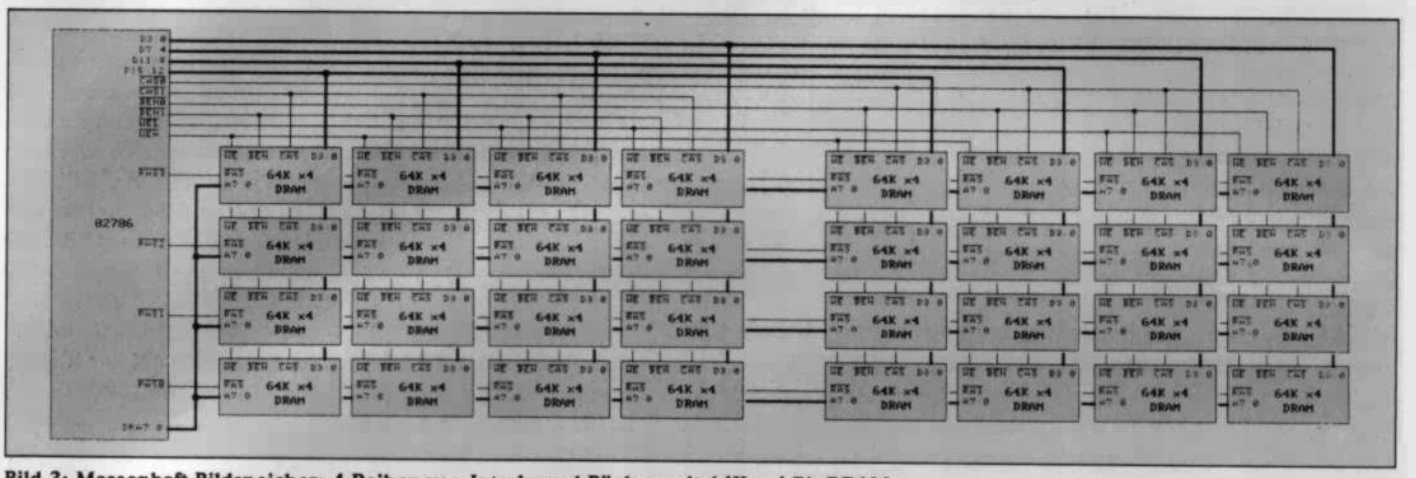


Bild 3: Massenhaft Bildspeicher: 4 Reihen von Interleaved Bänken mit 64K x 4 Bit DRAMs

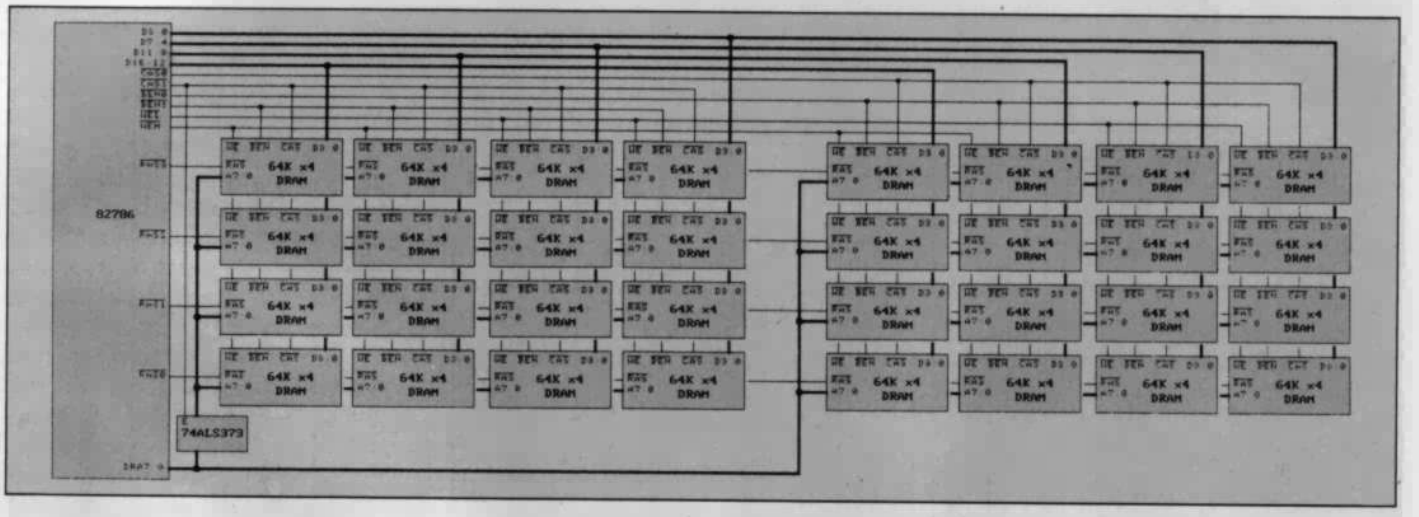
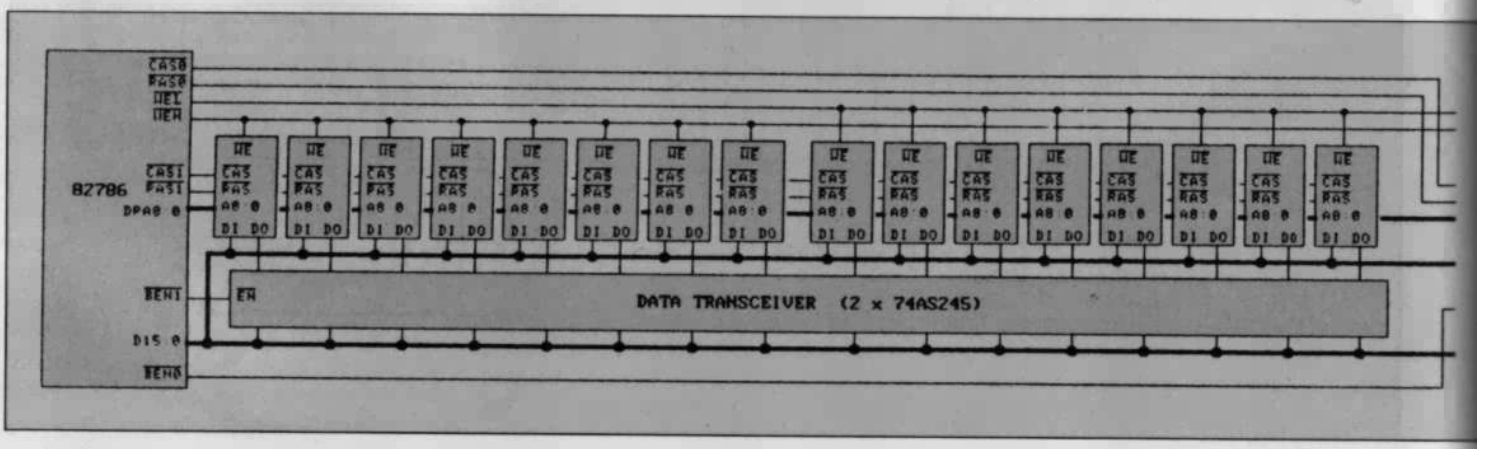


Bild 4: Bei Static Column DRAMs kommt zur Schaltung aus Bild 3 noch ein externer Zwischenspeicher (74 ALS 373) hinzu



ters wird durch das POR-Signal für eine längere Null-Phase des RESET-Signales gesorgt, dann folgt ein Impuls von ca. 4,5 µs. Dieser Impuls ist so kurz, um bei einem Reset während des Betriebes keine Daten im (dynamischen) Grafikspeicher zu verlieren. Ein Reset wird auch ausgelöst, wenn der GDP 82786 versucht, in den Master-Modus zu gehen (HREQ aktiv).
Im Master-Modus will der GDP 82786 einen DMA-Zugriff auf den Speicher der CPU 68000 durchführen, was von dieser einfachen Karte nicht unterstützt wird, da eine große Zahl zusätzlicher Adreßpuffer und Schaltglieder nötig wäre. Dies bedeutet aber kaum eine Einschränkung für den Betrieb der Karte, da der Grafikprozessor bis zu 4 MByte eigenen RAMs adressieren kann. Bei richtiger Programmierung wird das Signal HREQ niemals aktiv. Im Falle eines Fehlers muß der GDP 82786 zurückgesetzt werden, da es sonst zum „Aufhängen" des Grafikprozessors und eventuell des gesamten Computers führt.
Solange das RESET-Signal aktiv ist, wird über ein Und-Gatter die Leitung BHE auf logisch 0 gehalten. Dadurch wird der GDP auf asynschrones Busprotokoll eingestellt. Die Schaltung in Bild 1 ist für 1 MByte Video-RAM ausgelegt, allerdings sind nur 256 KByte eingezeichnet. Weitere RAM-Bausteine werden einfach parallelgeschaltet und statt über RAS0 über die Signale RAS1...RAS3 selektiert (→ Bild 4).
Die Adreßleitungen A20 und A21 des GDP sind fest mit Masse verbunden, da nur Adressen von 0...1 MByte erzeugt werden müssen.

Timing

Read-, Write- und Select-Signal lassen sich einfach von den Bussignalen der CPU 68000 ableiten. Zu beachten ist, daß das Ansprechen des GDP im asynchronen Modus vom RD- und WR-Signal kontrolliert wird, daher werden diese nur in Verbindung mit einem Chip-Selekt aktiviert. Das Signal SEN (Slave-Enable, Slave-Zugriffe sind alle normalen Buszyklen) zeigt das Beenden eines Zugriffes auf den GDP an. Aus diesem Signal kann daher sehr einfach das DTACK-Signal für die CPU 68000 sowie ein Steuersignal für die Datenlatches am Bus (74 ALS 646) gewonnen werden. Diese arbeiten bei Lesezugriffen des 68000 als Latches und bei Schreibzugriffen transparent.
Das DTACK-Signal wird über ein Flipflop (IC 14) erzeugt. Es ist nicht nötig, die Freigabe der Datenpuffer mit den Daten-Strobes zu verknüpfen, da die Ausgänge der Speicherbausteine bei Schreibzugriffen hochohmig geschaltet werden.
Wird die Grafikkarte in einem Steckplatz des mc-68000-Computers montiert, so müssen 2 MByte Adreßraum reserviert werden. Im ersten MByte wird der Bildspeicher selektiert. Die CPU kann auf den gesamten Bildspeicher transparent zugreifen, die Wartezeiten sind i.a. sehr gering, so daß sich eine kaum merkliche Verzögerung ergibt. Das zweite MByte wird in 4 Bereiche aufgeteilt:

1. I/O-Bereich des GDP 82786: Hier werden die internen Register des Grafikprozessors angesprochen
2. CLUT: Hier stehen die Register der Lookup-Table
3. FF: Ein Steuerflipflop zum Ändern der Punktfrequenz
4. Mouse: Ein Adreßbereich für eine Maus-Schnittstelle oder einen V.24-Baustein

Theoretisch bietet der Decoder (74 LS 138) noch weitere Bereiche, die man für andere Schaltungszusätze verwenden kann.
Bild 2 zeigt das Zeitverhalten des GDP 82786 bei asynchronen Buszugriffen auf seine Register oder den Grafikspeicher. Vergleicht man Bild 2 mit dem Verhalten des Prozessor 68000, so sieht man leicht, daß die fallende Flanke des SEN-Signales gut geeignet ist, um das DTACK-Signal zu erzeugen.

Speicher

Hinsichtlich des Grafikspeichers bietet der GDP 82786 sehr viele Möglichkeiten. Dazu vorab einige grundsätzliche Dinge: Standard-DRAMs (z.B. 41256 oder 41464) lassen sich neben der Einzel-Zugriffsart auch im Page-Modus betreiben. Beim Page-Modus wird im RAM über das RAS-Signal eine Reihe selektiert, anschließend werden in mehreren CAS-Zyklen beliebige Spalten dieser Reihe ausgelesen. Da beim Schreiben des Bildes in der Regel immer aufeinanderfolgende Speicherplätze benötigt werden, ist diese Betriebsart sehr gut geeignet. Im Page-Modus entfällt so für alle Zyklen bis auf den ersten das Anlegen der RAS-Adresse sowie die Precharge-Zeit, die das DRAM nach einem RAS-Impuls braucht. Die Zykluszeit verringert sich so auf die spezifizierte Zugriffszeit des RAM-Bausteins (z.B. 120 ns oder 150 ns). Intel definierte nun einen weiteren Modus, nämlich den Fast-Page-Modus. In dieser Betriebsart verringert sich die Zykluszeit nochmals (auf etwa 80 ns). Das DRAM muß dazu intern die selektierte Reihe komplett zwischenspeichern. Aus diesem Zwischenspeicher sind die einzelnen Spalten dann schnell abrufbar. Eine andere DRAM-Version unterstützt die sogenannte Static-Column-Adressierung: Statt der Reihenadresse wird hier die Spaltenadresse festgehalten, dazu ist beim Betrieb mit dem GDP 82786 ein externes 8-Bit-Latch (74 AS 374) notwendig (*Bild 3*).
Abhängig von den verwendeten Speichern ergibt sich ein bestimmter Datendurchsatz und damit Grenzen für die Auflösung. Wie man in der Schaltung sieht, verfügt der GDP nur über einen 16 Bit breiten Datenbus zum Speicher. Um durch diesen – für Bildspeicher sehr schmalen – Bus die gesamten Video-Informationen zu schleusen, bedient sich der GDP zweier Tricks:
1. Zwei Speicher-Bänke (mit je 16 Bit) werden zueinander phasenverschoben angesteuert, die Daten werden dann kurz

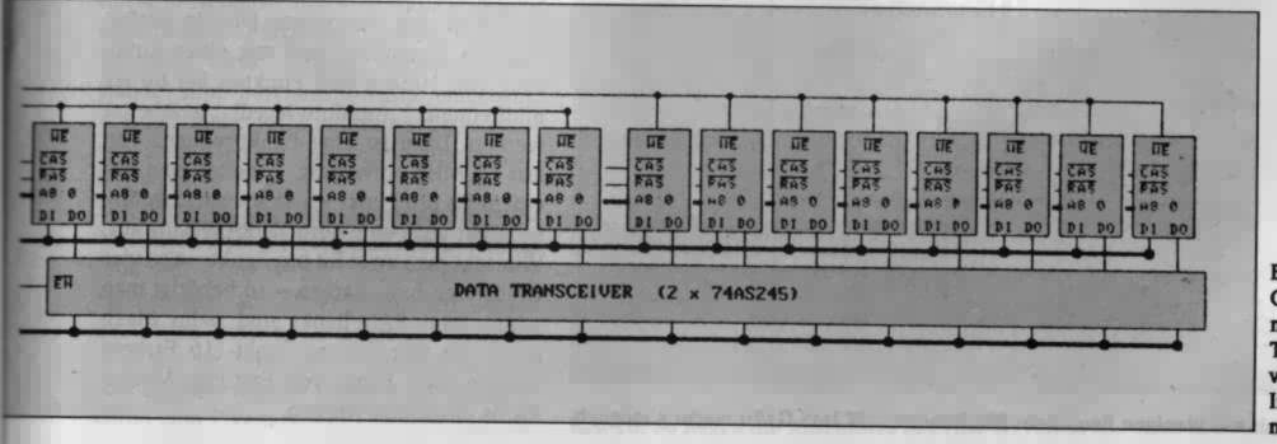


Bild 5: DRAMs ohne Output-Enable müssen über Tristate-Treiber geführt werden: Hier zwei Interleaved Bänke mit 256K x 1 Bit

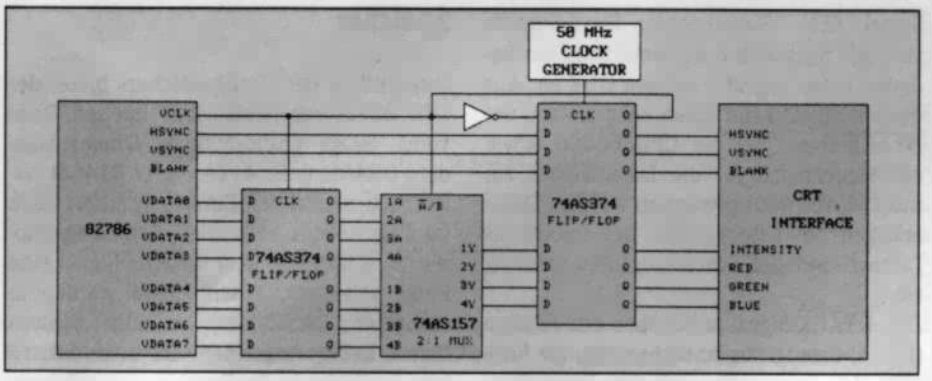


Bild 6: Die externe Taktverdopplung ermöglicht Punktfrequenzen bis 70 MHz

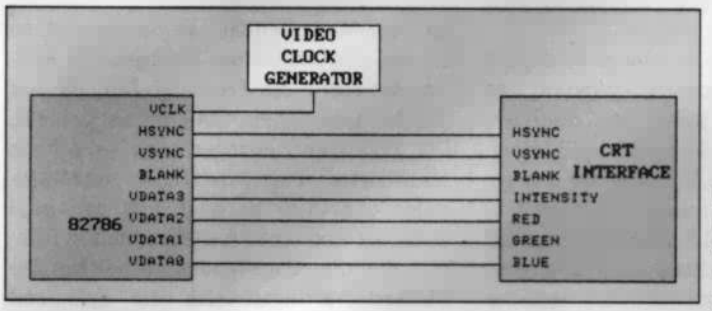


Bild 7: Einen Monitor mit TTL-Eingängen kann man direkt an den GDP 82786 anschließen ...

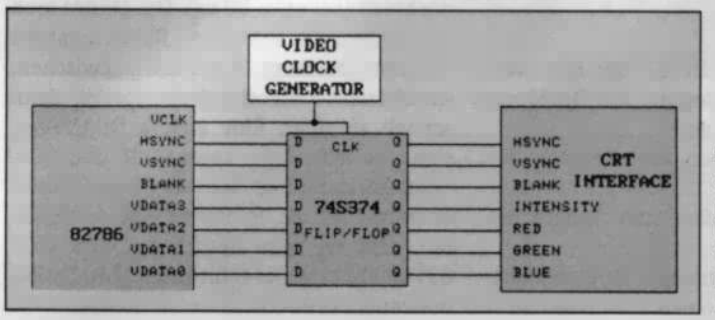


Bild 8: ... zum Schutz des teuren Grafikprozessors ist der Einsatz eines Treibers vor dem Monitor-Ausgang empfehlenswert

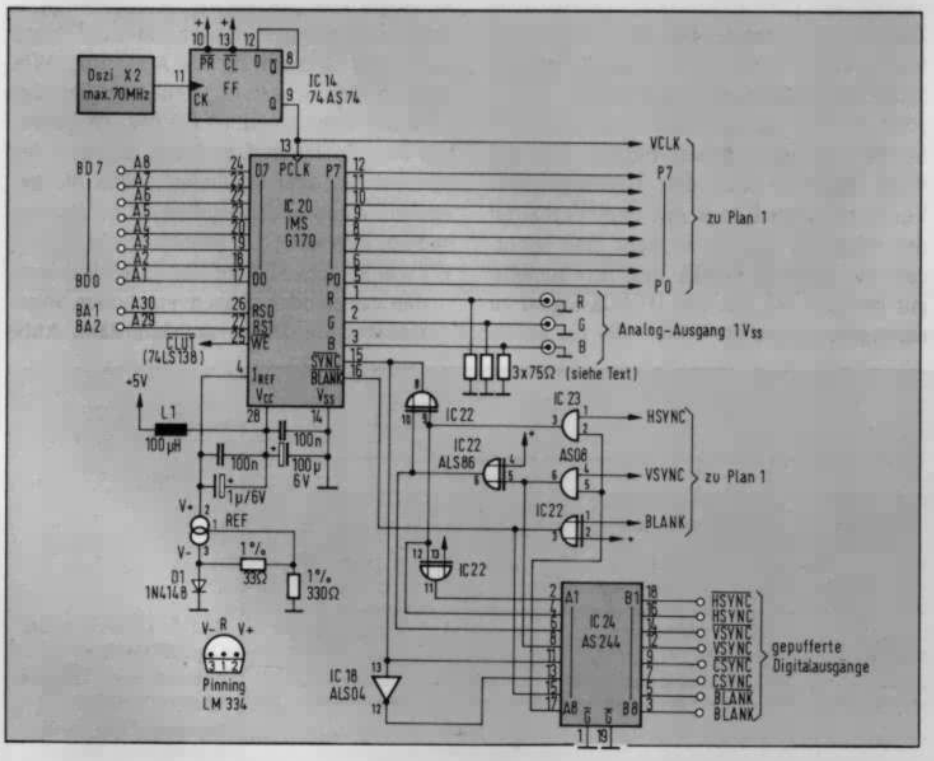


Bild 9: Viele Farben mit nur wenigen Bauteilen: Die Inmos-CLUT IMS G170 macht's möglich

nacheinander in die jeweilige Bank übertragen

2. Die Verwendung von Fast-Page-Modus-RAMs

Während der erste Punkt keine Mehrkosten erzwingt, da man meistens ohnehin relativ viel Bildspeicher verwenden möchte, gehen die Fast-Page-Modus-RAMs – oder alternativ Static-Column-RAMs – ins Geld.

Man kann sich beim Preis des Speichers also nach der Anwendung richten: Verwendet man eine RAM-Bank mit 16 Bit, so erhält man einen Datendurchsatz von 10 MByte/Sekunde. Mit zwei phasenverschobenen Speicher-Bänken (interleaved) oder Fast-Page-Modus-RAMs ist der Datendurchsatz auf 20 MByte pro Sekunde begrenzt. Beim Einsatz von zwei phasenverschobenen Bänken von Fast-Page-Modus-RAMs liegt der maximale Durchsatz bei knapp 40 MByte pro Sekunde. Die Art der verwendeten RAM-Bausteine hängt also von folgenden Faktoren ab:

1. gewünschte Auflösung bzw. maximale Punktfrequenz
2. gewünschte Bittiefe bzw. Farbzahl pro Punkt.

Dazu ein paar Beispiele: Für einen Standardbildschirm mit 640 x 300 Punkten Auflösung reicht eine Punktfrequenz von 16 MHz gut aus. Arbeitet man mit 8 Bit pro Punkt, so braucht man einen Datendurchsatz von 16 MByte pro Sekunde. Am billigsten ist daher die Verwendung von zwei phasenverschobenen Speicher-Bänken, wie sie Bild 1 zeigt. Braucht man nur 4 Bit pro Punkt, so kommt man auch mit einer einzelnen RAM-Bank aus.

Man muß bei der Bandbreite natürlich immer etwas Reserve lassen, da ja neben dem Display-Prozessor auch noch der Plotprozessor, der Window-Prozessor und die Haupt-CPU auf den Speicher zugreifen wollen. Läßt man zu wenig Luft, so verbleibt für diese nur die vertikale Rücklaufphase, da die horizontale für den Refresh der DRAMs verwendet wird. Das hätte merkliche Geschwindigkeitseinbußen zur Folge.

Zurück zu den Beispielen: Für ein professionelles Textsystem soll mit einer Auflösung von 1024 x 800 Punkten bei 60 Hz Bildfrequenz (noninterlaced) gearbeitet werden. Dazu ist eine Punktfrequenz von fast 70 MHz notwendig. Arbeitet man mit einem Bit pro Punkt – also schwarz/weiß -, so reicht noch eine normale RAM-Bank aus. Wünscht man zwei Bit pro Punkt – also vier Graustufen bzw. Farben – so benötigt man schon zwei RAM-Banks und beim Maximum von vier Bit pro Punkt (16 Farben) müssen zwei Banks von Fast-Page-Modus-RAMs eingesetzt werden.

Typische Anwendungen arbeiten mit Punktfrequenzen von 24...32 MHz. Es lassen sich damit Auflösungen von 640 x 400 ... 640 x 480 Punkten bei 50 oder 60 Hz Bildfrequenz erreichen. Hier kann man noch mit den preisgünstigen Multisync-Farbmonitoren arbeiten, die dem IBM-EGA-Standard genügen. Bei Verwendung von zwei phasenverschobenen RAM-Banks kann mit 4 Bit pro Punkt gearbeitet werden, bei Verwendung von zwei phasenverschobenen Banks mit Fast-Page-Modus-RAMs kann man das Maximum von 8 Bit pro Punkt, also 256 Farben pro Punkt einsetzen.

Der Grafikprozessor wird über ein Steuerregister informiert, wieviele Speicher angeschlossen und welche Betriebsarten möglich sind.

In Bild 1 ist die Beschaltung mit 8 DRAMs vom Typ 41464-12 eingezeichnet. Diese Bausteine sind zu 4 x 64 KBit organisiert, haben also jeweils eine Gesamtkapazität von 256 KBit. Der GDP 82786 erwartet von einem DRAM, daß es ein Output-Enable-Signal $\overline{OE}$ besitzt. Dies ist beispielsweise bei Bausteinen mit 1-Bit-Organisation (41256 o.ä.) nicht der Fall. Werden solche Chips verwendet, so müssen externe Puffer vorhanden sein, um die Ausgänge einzelner DRAMs hochohmig zu machen. Das ist nötig, um den phasenverschobenen Betrieb zu ermöglichen und bei Byte-Schreibzugriffen Kurzschlüsse zu vermeiden. *Bild 3...Bild 5* zeigen die Beschaltung des Grafikprozessors mit verschiedenen RAM-Bausteinen mit 256 KBit Kapazität. Bei Verwendung von 1 MBit-RAMs ergeben sich exakt die gleichen Schaltungen, lediglich eine Adreßleitung (DRA8) wird zusätzlich an die Speicherbausteine gelegt. Bild 4 zeigt auch die fehlenden Speicherchips der Grafikkarte. Prinzipiell ist der Aufbau der Karte mit 256 KByte, 512 KByte, 768 KByte und 1 MByte möglich. Wieviel Speicher man einsetzt, ist wohl hauptsächlich eine Kostenfrage und hängt von der Anwendung ab. Kleinere Systeme kommen sogar mit nur 4 RAM-Bausteinen aus, was 128 KByte entspricht. Der GDP kann bis zu 32 RAM-Bausteine direkt treiben, Impedanzwiderstände sind bereits im Chip eingebaut. Werden mehr RAM-Bausteine angeschlossen, so müssen alle Leitungen extern gepuffert werden.

Die Verwendung von Video-RAMs wird hier nicht beschrieben. Bei deren Einsatz muß auf punktweises Positionieren der Fenster und auf den eingebauten Cursor verzichten werden. Genauere Hinweise dazu geben die Datenblätter zum GDP 82786.

Alle Speicherzugriffe laufen zunächst asynchron zum Pixeltakt. Ein eigener Oszillator von 20 MHz taktet die Speicherzugriffe sowie den Plotprozessor des GDP 82786.

Die Video-Schnittstelle

Informationen aus dem Grafikspeicher werden immer im Page- oder Fast-Page-Modus in einen 32 Langworte fassenden FIFO-Speicher geladen. Aus diesem werden die Pixeldaten dann mit dem Video-Takt herausgeschoben. Der GDP 82786 gibt pro Takt 1...8 Bit auf seinen Videoausgängen aus. Die Zahl der Bits ist abhängig von der gewählten Bittiefe und kann für jedes Fenster einzeln gewählt werden (1, 2, 4 oder 8). Die Bits werden immer auf den niederwertigen Ausgängen ausgegeben, die ggf. unbenutzten höherwertigen Ausgänge lassen sich mit einer Default-Farbe besetzen.

Der Videotakt beträgt bis zu 35 MHz, womit der erreichbaren Auflösung eine obere Grenze gesetzt wird. Soll diese durchbrochen werden, so können die Videodaten extern durch Zusatzhardware gemultiplext werden, wodurch sich die doppelte (70 MHz), vierfache (140 MHz) oder sogar 8-fache (280 MHz) Pixelfrequenz erreichen läßt. Der GDP 82786 wird durch ein Steuerregister über einen dieser sogenannten 'Accelerated'-Modi informiert. Da die letzten beiden sich nur unter Verwendung teurer ECL-Technik realisieren lassen, ist hier nur die Betriebsart mit doppelter Takfrequenz angegeben *(Bild 6)*. Mit jeder Taktverdopplung halbiert sich natürlich die maximal erreichbare Bittiefe, bei 70 MHz sind so nur noch 16 Farben (≙ 4 Bit) pro Punkt möglich.

Die Digital-Ausgänge des GDP 82786 können direkt einen Monitor ansteuern *(Bild 7)*, besser ist jedoch das Zwischenschalten eines zusätzlichen Treibers *(Bild 8)*. Verwendet man hier einen TTL-Baustein der 'AS'- oder 'S'-Serie, so können 75Ω-Leitungen direkt gespeist werden. Das Blank-Signal ist im Prinzip überflüssig, da es indirekt bereits in den Pixeldaten enthalten ist. Die beste Darstellung in Bezug auf die Anzahl von Farben und Graustufen erreicht man natürlich mit einem analogen Bildsignal, wie es z.B. die CLUT IMS G170 *(Bild 9)* erzeugt. Hier stehen 256 Farben aus einer Palette von 256000 Farben bzw. mit 64 Graustufen zur Verfügung. Die CLUT erzeugt ein RS-170-kompatibles Ausgangssignal, das direkt in eine 75Ω-Leitung eingespeist wird. Ein Abschlußwiderstand von 75 Ω muß entweder an der CLUT oder am Monitor-Eingang angebracht sein! Synchron-Impulse und Blank-Signal sind be-

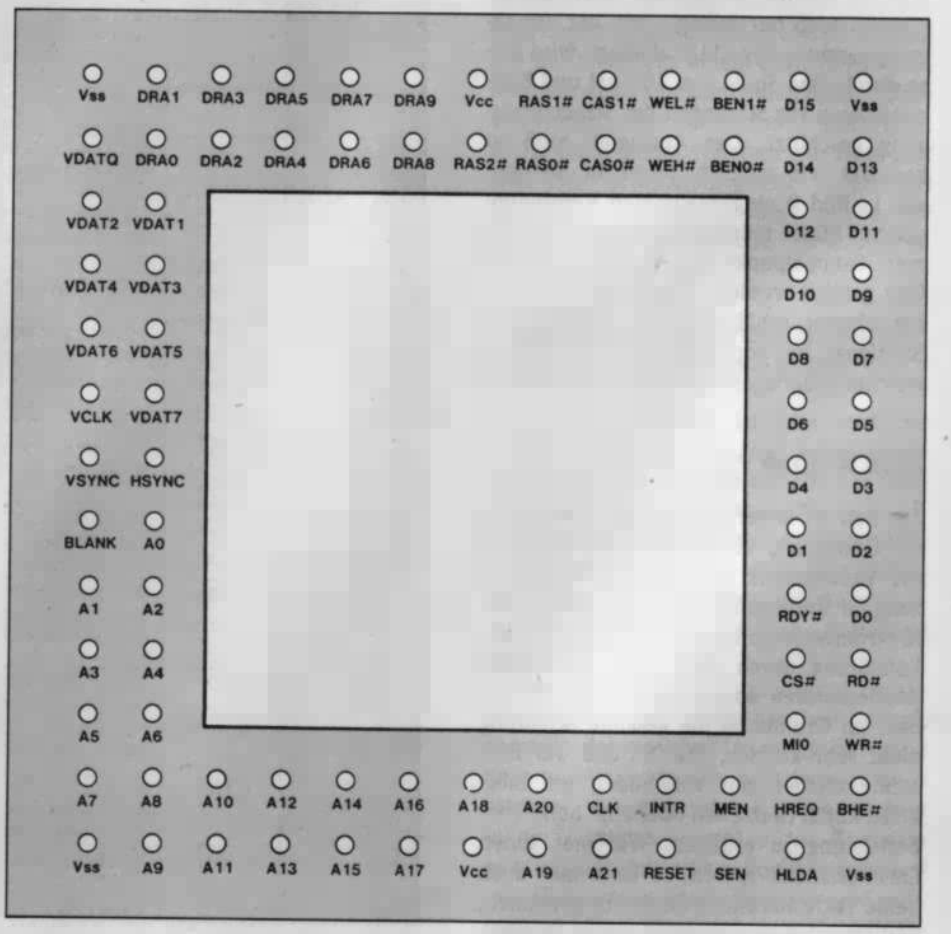


Bild 10: So liegen die Anschlüsse des GDP 82786 in der Sicht von Oben

reits in die Farbinformation eingemischt, können aber nochmals extra mit TTL-Pegel abgegriffen werden (das RS-170-Signal ist wie ein BAS-Signal aufgebaut). Man kann an die Clut natürlich auch einen Schwarzweiß-Monitor mit Analogeingang anschließen. Dazu wählt man einfach einen der drei Farbausgänge, die beiden anderen bleiben frei.

Für welche Lösung man sich auch entscheidet, wichtig ist die Verwendung von sauber verdrahteten Koaxialkabeln mit einer Impedanz von 75 Ω. Ein Flachkabel oder ungeschirmte Leitungen wären hier vollkommen fehl am Platz.

Entstörung

Die Clut ist sehr empfindlich gegenüber Schwankungen des Referenzstromes I_{ref}, über den die Digital-Analog-Wandler arbeiten. Aus diesem Grund wird zur Erzeugung des Referenzstromes eine Präzisions-Konstantstromquelle verwendet – keine Angst, sie kostet nur knapp 3 Mark. Zwei Widerstände stellen den Referenzstrom ein, sie sollten deshalb möglichst temparaturstabil sein. Eine Siliziumdiode 1N4148 kompensiert Temparaturschwankungen des LM 334. Da nun die Stromquelle selbst wieder – wenn auch nur gering – von der Versorgungspannung der CLUT abhängt, wird diese durch eine Spule von 100 µH und Kondensatoren vor Störungen der Stromversorgung geschützt. Der Analogteil muß so kompakt wie möglich aufgebaut werden, alle in Bild 9 eingezeichneten Verbindungen zur Masse müssen auf einen gemeinsamen Sammelpunkt am V_{ss}-Anschluß der Clut geführt werden. Wenn man hier sauber arbeitet, erhält man ein einwandfreies Bildsignal, frei von Störungen oder irgendwelchen Überlagerungen.

Zum Aufbau

Zur Zeit ist leider kein fertiges Layout für die Grafikkarte vorhanden, man ist daher auf Fädeltechnik angewiesen. Voraussetzung für störungsfreie Grafik ist eine saubere Stromversorgung (dicke Leitungen), gute Entstörung durch Elkos und keramische Kondensatoren sowie ein kompakter Aufbau. Im Grunde ist die gesamte Schaltung nicht sehr kritisch, die 'S'- und 'AS'-Bausteine werden nur verwendet, um möglichst kurze Laufzeiten oder eine hohe Treiberleistung zu erhalten. Will man einen Quarzoszillator sparen, so kann man auch beide Takte aus einem Oszillator gewinnen, verliert dabei aber natürlich etwas Flexibilität.

(Fortsetzung folgt)

Symbol	Type	Description
A21–0	I/O	**ADDRESS LINES FOR THE LOCAL BUS:** Normally inputs for Slave Mode accesses of the 82786 supported DRAM array or internal memory or I/O mapped registers. Driven by the 82786, when it is the Local Bus Master.
D15–0	I/O	**DATA BUS:** For the 82786 DRAM array and the Local Bus.
BHE#	I/O	**BUS HIGH ENABLE:** An input of the 82786 Slave Interface: driven LOW by the 82786 when it is Local Bus Master. Determines asynchronous vs. synchronous operation for RD#, WR# and HLDA inputs at the falling (trailing) edge of RESET. A HIGH state selects synchronous operation.
RD#	I/O	**READ STROBE:** An input of the 82786 Slave Interface: driven by the 82786 when it is Local Bus Master. Asynchronous vs. synchronous input determined by state of BHE# pin at falling RESET.
WR#	I/O	**WRITE STROBE:** An input of the 82786 Slave Interface: driven by the 82786 when it is Local Bus Master. Asynchronous vs. synchronous input determined by state of BHE# pin at falling RESET.
MIO	I/O	**MEMORY / I/O INDICATION:** An input of the 82786 Slave Interface: driven HIGH by the 82786 when it is the Local Bus Master. Selects 286 Status or Command Mode vs. 8086/186 Status Mode of the 82786 Slave Interface at the falling (trailing) edge of RESET. A LOW state selects the 286 Status or Command Mode.
CS#	I	**CHIP SELECT:** Slave Interface input qualifying the access.
MEN	O	**MASTER ENABLE:** Driven HIGH when the 82786 is in control of the Local Bus, (i.e. HLDA received in response to a 82786 HREQ). Used to steer the data path and select source of bus cycle status commands.
SEN	O	**SLAVE ENABLE:** Driven HIGH when 82786 is executing a Slave bus cycle for an external master on the Local Bus. Used to enable the data path and as a READY indication to the Local Bus Master.
READY#	I	**SYNCHRONOUS INPUT:** To the 82786 when executing Local Bus cycles. Identical to 80286 timing.
HREQ	O	**HOLD REQUEST:** Driven HIGH by the 82786 when an access is being made to the Local Bus by the Display or Graphics Processors. Remains HIGH until the 82786 no longer needs the Local Bus.
HLDA	I	**HOLD ACKNOWLEDGE:** Input in response to a HREQ output. Asynchronous vs. synchronous input determined by state of BHE# pin at falling RESET.
INTR	O	**INTERRUPT:** The logical OR of a Graphics Processor and Display Processor interrupt. Cleared with an access to the BIU Interrupt Register.
CAS0#	O	**COLUMN ADDRESS STROBE 0:** Drives the CAS inputs of the even word DRAM bank if interleaved: identical to CAS1# if non-interleaved DRAM. Capable of driving 16 DRAM CAS inputs.
CAS1#	O	**COLUMN ADDRESS STROBE 1:** Drives the CAS inputs of the odd word DRAM bank if interleaved; identical to CAS0# if non-interleaved DRAM. Capable of driving 16 DRAM CAS inputs.
RAS2–0#	O	**ROW ADDRESS STROBE:** Drives the RAS input pins of up to 16 DRAMs. Drives the first three rows of both banks of DRAM.
DRA9/ RAS3#	O	**MULTIPLEXED MOST SIGNIFICANT DRAM ADDRESS LINE AND RAS3#:** Support of 1Mb DRAMs requires DRA9. When 1Mb DRAMs are used, four rows of DRAMs cannot be supported (RAS3# unnecessary) due to 82786 addressing limit of 4 Mbytes being exceeded.
WEL#	O	**WRITE ENABLE LOW BYTE:** Active LOW strobe to the lower order byte of DRAM.
WEH#	O	**WRITE ENABLE HIGH BYTE:** Active LOW strobe to the higher order byte of DRAM.
DRA8–0	O	**MUTIPLEXED DRAM ADDRESS:** DRAM row and column address are multiplexed on these lines. Capable of driving 32 DRAMs without buffers.
BEN1–0#	O	**BANK ENABLE 1 AND 0:** Enables the output of the DRAM array on to the 82786 data bus (D15–0). BEN1# controls Bank 1. BEN0# controls Bank 0.
BLANK	I/O	**OUTPUT USED TO BLANK THE DISPLAY AT PARTICULAR POSITIONS ON THE SCREEN:** May also be configured as inputs to allow the 82786 to be synchronized with external sources.
VDATA7–0	O	**VIDEO DATA OUTPUT.**
VCLK	I	**VIDEO CLOCK INPUT:** used to drive the display section of the 82786. Its maximum frequency is 25 MHz.
HSYNC/ WST0	I/O	**HORIZONTAL SYNC:** Window status may be multiplexed on this pin. Can also be configured as input to allow the 82786 to be synchronized with external sources. Even as input, window status still output when BLANK is low.
VSYNC/ WST1	I/O	**VERTICAL SYNC:** Window status can be multiplexed on this pin. Can also be configured as input to allow the 82786 to be synchronized with external sources. Even as input, window status still output when BLANK is low.
RESET	I	**RESET INPUT:** internally synchronized. Halts all activity on the 82786 and brings it to defined state. The leading edge of RESET synchronizes the clock to PH1. The trailing edge latches the state of BHE# and MIO to establish the type of Slave Interface. It also latches RD# and WR# to set certain test modes.
CLK	I	**DOUBLE FREQUENCY CLOCK OUTPUT:** Clock input to which pin timings are referenced. 50% duty cycle.
V_{SS}, V_{CC}		**4 V_{SS} AND 2 V_{CC} PINS.**

Bild 11: Die Bedeutung der Signale des GDP 82786

Klaus Manhart

Prolog und Logik

Der Beweismechanismus von Prolog und seine logischen Grundlagen (Teil 2)

Prolog ist die derzeit höchste Programmiersprache, die ganz allgemein für die Bearbeitung nicht-numerischer Problemstellungen eingesetzt werden kann. Im zweiten und letzten Teil werden Clausenlogik, Hornclauseln, Syntax und die Resolutionslogik behandelt.

In der Clausenlogik werden die prädikatenlogischen Formeln in eine sogenannte Normalform umgewandelt. Hier gibt es nur mehr negierte und nicht-negierte Atomformeln, die ausschließlich mit „und" und „oder" verknüpft sind. Jede prädikatenlogische Standardformel läßt sich in einem algorithmischen Prozeß in Clausenlogik überführen.

Als Beispiel die Formel
(F0) ∀X: ∀Y: p(X,Y) & q(Y) =>
∃Z: r(Z) & p(X,Z)

Sie kann als prädikatenlogische Repräsentation der folgenden Regel aufgefaßt werden: Für alle X,Y gilt: Wenn X Kind von Y ist und Y männlich ist, dann gibt es ein Z, so daß Z weiblich und X Kind von Z ist, oder etwas natürlicher formuliert:
Zu jedem Kind eines Vaters gibt es eine Mutter (statt der „sprechenden" Prädikate kind_von etc. verwenden wir hier kurze Ausdrücke).
Die Umwandlung prädikatenlogischer Formeln in Formeln der Clausenlogik geschieht mit folgenden Schritten:

– Ersetze Implikationen und Äquivalenzen unter Benutzung der Gesetze:
F <=> G ist äquivalent zu
F => G & G => F;
F => G ist äquivalent zu ¬F ∨ G.

(Die Beweise für diese sowie die in den folgenden Schritten angegebenen Äquivalenzen sind trivial.)
Aus der obigen Formel (F0) wird somit
(F1) ∀X: ∀Y: ¬(p(X,Y) & q(Y)) ∨
∃Z: r(Z) & p(X,Z).

– Schiebe Negationssymbole möglichst weit nach innen unter Nutzung der Gesetze:
¬(¬F) äquivalent zu F;
¬(F & G) äquivalent zu ¬F ∨ ¬G;
¬∀X: p(X) äquivalent zu ∃X: ¬p(X);
¬∃X: p(X) äquivalent zu ∀X: ¬p(X).
Somit ergibt sich (F1) zu
(F2) ∀X: ∀Y: (¬p(X,Y) ∨ ¬q(Y)) ∨
∃Z: r(Z) & p(X,Z).

– Eliminiere Existenzquantoren durch Skolemisierung:
Existenzquantoren sind für automatische Theorembeweiser hinderlich. Nach dem Logiker Skolem lassen sich „erfüllbarkeitsäquivalente" existenzquantorfreie Formeln herstellen durch Ersetzung der existenzquantifizierten Variablen mit einer neuen Konstanten, oder, falls die Variable im Bereich eines Allquantors liegt, durch Ersetzung mit (Skolem-)Funktionen. Die Argumente von Skolemfunktionen bilden dabei sämtliche allquantifizierten Variablen, in deren Bereich die existenzquantifizierten liegen.
Beispielsweise müßte die Formel
∀X: ∃Y: p(X,Y) transferiert werden zu:

∀X: p(X,f(X)).
Aus (F2) wird somit
(F3) ∀X: ∀Y: (¬p(X,Y) ∨ ¬q(Y)) ∨
(r(g(X,Y)) & p(X,g(X,Y))).

– Erzeuge die Pränexform:
Da nun in der Formel nur mehr Allquantoren vorkommen, können diese gedanklich nach vorne geschoben und weggelassen werden. Alle nun noch vorkommenden Variablen bleiben aber (implizit) allquantifiziert:

(F4) (¬p(X,Y) ∨ ¬q(Y)) ∨ (r(g(X,Y)) &
p(X,g(X,Y)).

– Konvertiere die Formel in eine Konjunktion von Disjunktionen (konjunktive Normalform) mit Hilfe der Distributivgesetze:

(F & G) ∨ H ist äquivalent zu (F ∨ H) &
(G ∨ H);
F ∨ (G & H) ist äquivalent zu (F ∨ G) &
(F ∨ H).
(F5) ((¬p(X,Y) ∨ ¬q(Y)) ∨ r(g(X,Y))) &
((¬p(X,Y) ∨ ¬q(Y)) ∨ p(X,g(X,Y)).

– Erzeuge die endgültige Clausenform:
Es liegen jetzt nur mehr Formeln vor, deren Hauptoperator das logische „und" ist. Man bezeichnet jedes einzelne Konjunktionsglied als eine Clause. Innerhalb der Clausen erscheinen nur mehr disjunktiv verknüpfte Formeln. Da bei „v" als einzigem Verknüpfungsoperator Klammerung und Reihenfolge innerhalb einer Clause irrelevant sind, werden Klammern weggelassen und die Formeln in einer Clause können zu einer (implizit disjunktiv verknüpften) Menge zusammengefaßt werden. Alle Clausen zusammen bilden schließlich eine (implizit konjunktiv verknüpfte) Menge von Clausen ((F6) siehe *Bild 1*).
Die Elemente einer Clause heißen Literale, negierte Elemente sind negative, unnegierte positive Literale. Eine Clause ist also eine Menge von positiven oder negativen Literalen.
Prolog liegt eine spezielle Variante von Clausen zugrunde, sog. Horn-Clausen (nach dem Logiker Horn). Hornclausen sind Clausen mit höchstens einem positiven Literal. Beispielsweise sind die beiden obigen Clausen Hornclausen, nicht aber {¬p(X), q(X), s(X)}. Nach einem Theorem von Horn läßt sich jede prädikatenlogische Formel als Hornclause darstellen; diese bilden die grundlegende syntaktische Struktur von Prolog.

Zusammenhang zwischen Hornclausen und Prolog-Syntax

Ein Prolog-Programm ist eine endliche Menge von Hornclausen, die in Prolog-Syntax geschrieben werden; anstelle der Mengennotation tritt die Satznotation:
Besteht eine Clause aus genau einem positiven Literal, so wird nur diese geschrieben und mit einem Punkt abgeschlossen. Clausen mit genau einem positiven Literal sind Fakten.
Besteht eine Clause aus mehreren negativen Literalen und einem positiven Literal, so steht das positive Literal immer ganz links und wird von den negativen durch :- getrennt, die negativen Literale werden (ohne Negationszeichen) durch Kommata getrennt und mit einem Punkt abgeschlossen. Clausen mit einem positiven und mindestens einem negativen Literal sind Regeln.

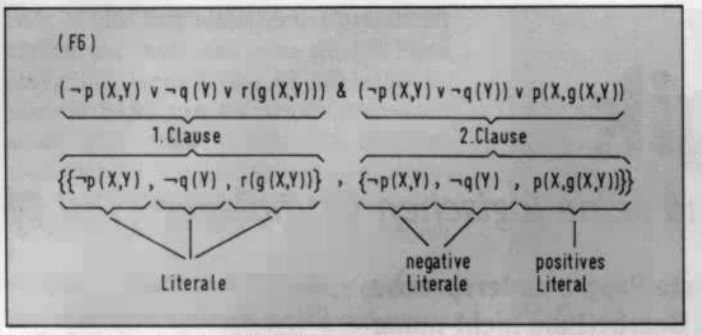


Bild 1. Eine implizit konjunktiv verknüpfte Menge von Clausen

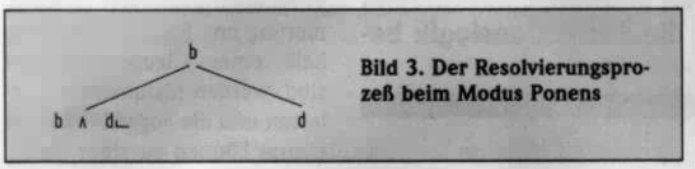


Bild 3. Der Resolvierungsprozeß beim Modus Ponens

1)	Standard-Logik:	p
	Clausenform:	{p}
	PROLOG-Notation:	p.
2)	Standard-Logik:	p ⇒ q
		(äquivalent zu: q v ¬p)
	Clausenform:	{q, ¬p}
	PROLOG-Notation:	q :- p.
3)	Standard-Logik:	∀X: p(X) & r(X) ⇒ q(X)
		(äquiv. zu: ∀X: (¬p(X) v q(X) v ¬r(X)))
	Clausenform:	{¬p(X), q(X), ¬r(X)}
	PROLOG-Notation:	q(X) :- p(X), r(X).
4)	Standard-Logik:	¬∃X: p(X)
		(äquiv. zu: ∀X: ¬p(X))
	Clausenform:	{¬p(X)}
	PROLOG-Notation:	?- p(X)

Bild 2. Gegenüberstellung von Standard-Logik, Clausenform und Prolog-Notation an einigen Beispielen

Als letzte Möglichkeit verbleiben schließlich noch Clausen mit ausschließlich negativen Literalen. Clausen mit negativen Literalen werden mit ?- beginnend und durch Kommata getrennt geschrieben. Clausen mit ausschließlich negativen Literalen sind – aus Gründen, die unten dargelegt werden – Fragen. Normalerweise ist dies eine einelementige Menge. Clausen mit mehreren negativen Literalen sind – wie man sich leicht überlegen kann – konjunktiv verknüpfte Fragen. Einige Beispiele zeigt *Bild 2.*

Aus (2) und (3) des Bildes 2 ist ersichtlich, daß die Prolog-(Satz-)Notation eine Horn-Clause in eine umgekehrt geschriebene Implikation umwandelt, so daß die Prolog-Notation immer als umgekehrter Wenn-Dann-Satz gelesen werden kann. Beispiel (4) zeigt, daß eine Prolog-Frage einem negierten Literal entspricht bzw. in der Standardlogik einer negierten Existenzformel (es gibt kein X, so daß X die Eigenschaft p hat). Der Grund hierfür wird sogleich deutlich.

Es sei darauf hingewiesen, daß man bei der praktischen Arbeit mit Prolog meist nicht so vorgeht, eine Wissensbasis prädikatenlogisch zu erstellen und sie in dem beschriebenen Prozeß in Prolog-Hornclausen zu überführen. Die Wissensbasis läßt sich meist direkt in Form von Prolog-Sätzen schreiben. Oft lassen sich durch die Einführung spezifischer „ad-hoc-Regeln" auch Skolemfunktionen vermeiden.

Der Beweismechanismus von Prolog

Nun zum zentralen Kern von Prolog und dem eigentlichen Ziel dieses Aufsatzes, der Vorstellung der Resolutionslogik. Der Beweismechanismus von Prolog und der meisten heutigen Deduktionsprogramme beruht auf dem Resolutionsprinzip von Robinson (1965):

Ist eine Interpretation I ein Modell für eine Clausenmenge C = {A,B}, so daß es ein Literal L gibt mit L∈A und L∈B, dann ist I auch Modell für C' = {(A\{L}) ∪ (B\{¬L})}. („\" ist das Operationszeichen für Differenzmengenbildung.)

Ist z. B. C = {{p},{q,¬p}} wahr, dann sind die Voraussetzungen des obigen Satzes erfüllt, da p∈{p} und ¬p∈{q,¬p}, und somit ist auch C' = {{p}\{p} ∪ {q,¬p}\{¬p}} = {q} wahr.

Der Schluß

{p}
{q,¬p}
{q}

ist natürlich nichts anderes als der klassische Modus Ponens, da {q,¬p} aussagenlogisch p => q ist.

Man nennt q die Resolvente und stellt den Resolvierungsprozeß zweckmäßigerweise dar, wie es *Bild 3* zeigt.

Die Ableitungsmethode in der Resolutionslogik beruht auf einem Widerspruchsbeweis: Man negiert die zu beweisende Formel – also das Theorem – und zeigt, daß es kein Modell gibt, das alle Formeln erfüllt. Der Hintergrund ist folgender:

Die Axiome sind als wahr vorausgesetzt. Folgt das Theorem aus den Axiomen, so müßte die Negation des Theorems unverträglich mit den Axiomen sein. Man sucht also einen logischen Konflikt zwischen dem negierten Theorem und den Axiomen. Semantisch ausgedrückt: Gibt es kein Modell, das alle Formeln erfüllt, so muß die negierte Formel falsch sein, die unnegierte mithin wahr. Konkret bedeutet dies nun, eine Clause herzuleiten, die nie erfüllbar ist und dies ist die leere Clause [ ]: eine Clausenmenge M ist nämlich widerspruchsvoll genau dann, wenn aus M die leere Clause ableitbar ist.

Um eine Formel s zu beweisen, ist diese also zu negieren, in die übrige Clausenmenge einzufügen und die leere Clause herzuleiten. Aus diesem Grund ist eine Prolog-Frage implizit eine negierte Clause, die mit den anderen Clausen aus der Datenbasis resolviert wird.

Prolog bedient sich beim Resolvieren der linearen Input-Resolution, nach der das Beweisziel – also das negierte Literal – resolviert wird mit einer Clause aus der Wissensbasis, die Resolvente dieses Prozesses wieder resolviert wird mit einer Clause aus der Wissensbasis usw. Die Auswahl der Resolutionskandidaten in der Datenbasis erfolgt dabei nach den Strategien:

– der Rumpf von Regeln wird von links nach rechts bearbeitet;
– Clausen werden in der Reihenfolge von oben nach unten durchsucht.

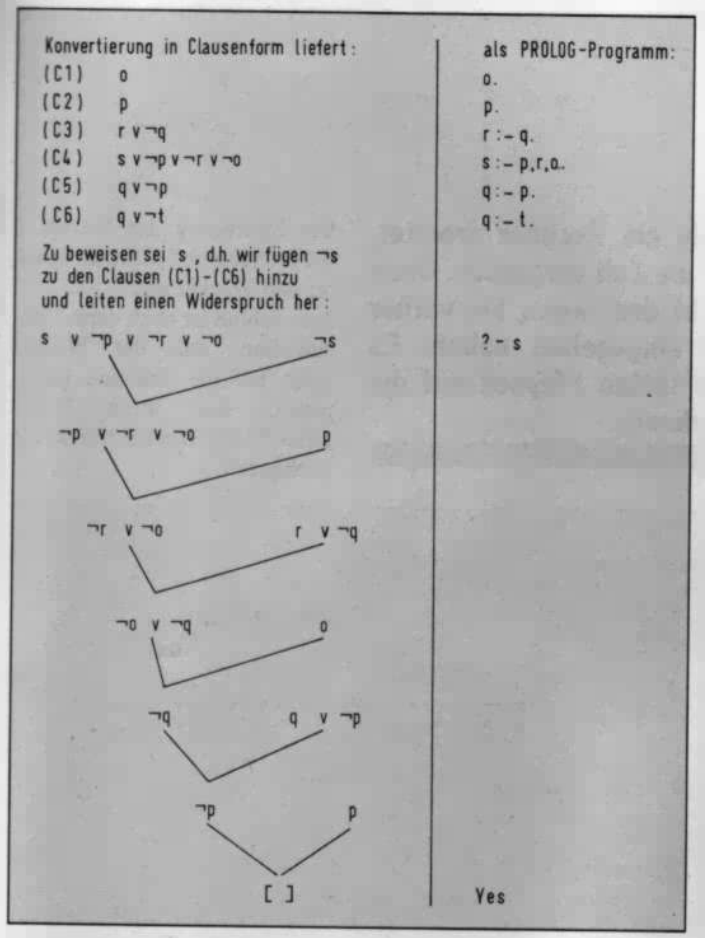
Angenommen, es seien folgende aussagenlogische Formeln gegeben:

(A1) o
(A2) p
(A3) q => r
(A4) p & r & o => s
(A4) p v t => q

Bild 4 zeigt die Konvertierung in Clausenform, das zugehörige Prolog-Programm und den Resolutionsbeweis für s. Aus diesem Beispiel ist unmittelbar nachprüfbar, daß die Anordnung der Clausen die Effizienz einer Lösung beeinflussen kann.

Bei prädikatenlogischen Formeln kommen nun zusätzlich Variable vor, die je nach Bindung einen Widerspruch erzeugen können oder nicht. Beispielsweise läßt sich bei den Clausen

praed(X).
praed(Y).



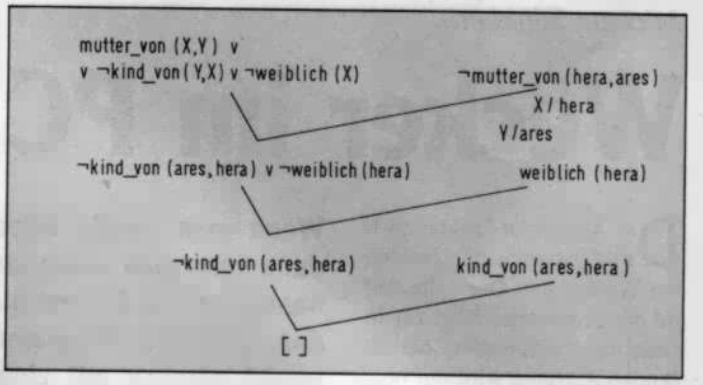


Bild 5. Prädikatenlogischer Resolutionsbeweis bei Ja/Nein-Fragen

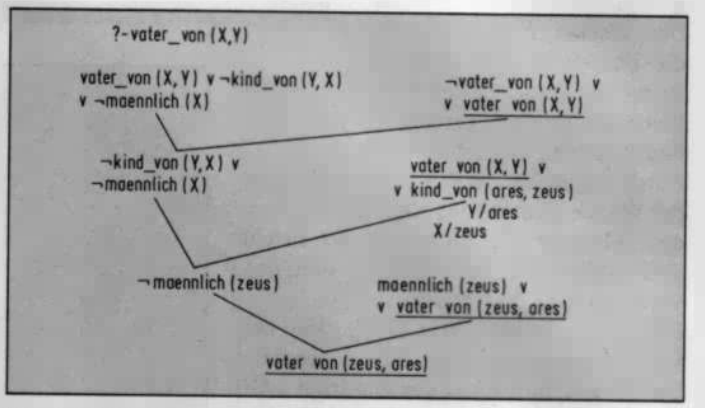


Bild 6. Prädikatenlogischer Resolutionsbeweis bei Ergänzungsfragen

◁ **Bild 4. Ein Resolutionsbeweis in der Aussagenlogik**

für die Belegung X=Y=a die leere Clause herleiten, nicht jedoch für X=a, Y=b.

Die theoretische Basis für das Resolvieren in der Prädikatenlogik ist das Theorem von Herbrand: es besagt, daß bei der Herleitung der leeren Clause die Interpretation über einen eingeschränkten Bereich, das Herbrand-Universum, ausreichend ist. Hier kommt nun der Unifikationsalgorithmus mit ins Spiel. Der Unifikationsalgorithmus ist ein Matching-Prozess, der 2 Literale vergleicht und prüft, ob es eine Variablensubstitution gibt, bei der die Formeln identisch werden. In beiden Literalen müssen natürlich das Prädikatsymbol und die Zahl der Argumentstellen übereinstimmen. Die Substitutionsregeln sind einfach: Variable können andere Variable, Konstante oder Funktionen ersetzen, sofern die Funktion nicht eben diese Variable als Argument besitzt (ansonsten: Gefahr von Endlossubstitutionen!). Ohne auf weitere Details einzugehen, sei gleich das Familienrelationenbeispiel präsentiert; diesmal soll zur Abwechslung die griechische Mythologie herhalten:

```
kind_von(ares,zeus).
kind_von(ares,hera).
maennlich(zeus).
maennlich(ares).
weiblich(hera).
mutter_von(X,Y) :-
kind_von(Y,X), weiblich(X).
vater_von(X,Y) :-
kind_von(Y,X), maennlich(X).
?- mutter_von(hera,ares).
```

Bild 5 zeigt den prädikatenlogischen Resolutionsbeweis (X/hera bedeutet, daß X mit hera unifiziert wird). Bei Ergänzungsfragen funktioniert der Beweisalgorithmus mit einer kleinen Modifikation: man fügt die ursprüngliche Frage ihrer Negation hinzu und markiert sie, so daß sie vom Resolutionsalgorithmus nicht verwendet wird. Ihre Argumente werden aber durch die stattfindenden Resolutionen gebunden. Das Ende des Verfahrens ist dann erreicht, wenn nur noch genau diese markierte Clause übrigbleibt; die Argumentbindungen stellen dann das Ergebnis dar: *Bild 6* zeigt einen prädikatenlogischen Resolutionsbeweis bei Ergänzungsfragen.

Abschließend sei noch darauf hingewiesen, daß der Resolutionskalkül die beiden wichtigsten Eigenschaften für Logikkalküle erfüllt: zum einen ist die Resolution korrekt, d. h. es werden keine falschen Aussagen hergeleitet, zum anderen ist sie vollständig bezüglich der Widerlegung, d. h. sind Clausen inkonsistent, so ist es immer möglich, die leere Clause abzuleiten.

Bei einem in eine Programmiersprache umgesetzten Resolutionskalkül wie Prolog ist die Vollständigkeit aber durch die informatische Interpretation nicht immer gegeben, insbesondere dann nicht, wenn ein Beweiskandidat nicht terminiert. Dies ist z. B. bei dem folgenden abschließenden Programm der Fall, bei dem durch die Auswahlstrategie für geeignete Resolventen eine Endlosschleife entsteht:

```
p(X) :- q(X).
q(X) :- p(X).
q(a).
?- p(X)
```

Um p(X) zu beweisen, ist nämlich q(X) zu beweisen. q(X) könnte zwar mit X=a bewiesen werden, steht aber hinter der Regel q(X) :- p(X), so daß diese zuerst aufgerufen wird. Um q(X) zu beweisen ist aber wiederum p(X) zu beweisen usw. Das Programm erreicht aufgrund der Anordnung der Clausen also nie den Kandidaten q(a) und gerät in eine Endlosschleife. Das Problem läßt sich zwar leicht beheben, indem man q(a) an erster oder zweiter Stelle setzt, zeigt aber auch, daß theoretisch herleitbare Lösungen in bestimmten Fällen nicht ermittelt werden können. Beim praktischen Programmieren können solche Probleme i.a. vermieden werden, indem Fakten grundsätzlich vor Regeln aufgeführt werden.

Die Literaturliste befindet sich aus Platzgründen auf S. 3.

Michael Senkbeil

Wecker im PC

Wenn man richtig intensiv am Rechner arbeitet, dann kann man schon mal die Zeit vergessen. Ohne unangenehme Folgen bleibt das, wenn Sie vorher das folgende Programm eingegeben haben: Es macht nämlich mit einem lauten Piepsen auf die vergessene Uhrzeit aufmerksam.

Das Assembler-Programm in *Bild 1* stellt einen residenten Wecker dar. Der erste Aufruf des Programms führt zur Installation im Speicher; bei folgenden Aufrufen wird nur noch dieses resident installierte Programm bedient.

Der Wecker fragt regelmäßig die Uhrzeit ab und vergleicht sie mit der gespeicherten Weck-Zeit. Ist die Weck-Zeit erreicht, erzeugt das Programm fünf Piepstöne und schaltet sich ab.

Durch den Befehl „WECKER SS:MM" wird der Wecker aktiviert (SS steht für die Stunden, MM für die Minuten), der Befehl „WECKER 17:00" beispielsweise stellt den Wecker auf 17.00 Uhr ein.

Durch den Befehl „WECKER" kann der Wecker jederzeit ausgeschaltet werden. Das Programm benutzt den Interrupt 1AH zur Zeitabfrage, so daß es auf jedem PC und AT ohne Änderungen lauffähig sein sollte. Probleme gibt es lediglich mit einigen Kompatiblen, die im BIOS einen geänderten Vektor auf die eigene Hardware-Uhr haben (z. B. Siemens PC-D, Olivetti M-24): Das Programm fragt die Software-Uhr ab, die aber – wegen des geänderten Vektors – nie gestellt wurde und daher die Uhrzeit seit Einschalten des Systems liefert. Daher muß in diesem Fall die Zeit direkt über den Uhrenport abgefragt werden. In mc 5/87, Seite 86 wird beschrieben, wie das auf dem Siemens PC-D funktioniert.

Die Turbo-Pascal-Prozedur in *Bild 2* ermöglicht die Benutzung des Programmes in Turbo-Pascal, nachdem es zuvor durch den Befehl „WECKER" im Speicher installiert worden ist. Es kann natürlich auch von anderen Hochsprachen verwendet werden. Die Speicheradresse des Weckers wird durch Abfragen der Speicherstelle 0:72 ermittelt, die die Segmentadresse der Routine im LSB/MSB-Format enthält. Nachdem die Segmentadresse ermittelt worden ist, müssen die ersten 6 Bytes auf den Inhalt „WEKKER" überprüft werden. Danach kann die Weckzeit geändert werden. Das Segment des Weckers hat folgenden Inhalt:

Offset 00: ASCII-Code „WECKER"
Offset 06: Minutenangabe der Weck-Zeit
Offset 07: Stundenangabe der Weck-Zeit

Die Stunden und Minuten der Weck-Zeit brauchen nur eingetragen zu werden. Die Stunden sollten zuerst eingetragen werden.

Die Eintragung des Wertes 0 bei Stunden und Minuten schaltet den Wecker aus.

Zum Schluß ist noch darauf hinzuweisen, daß der Wecker nicht auf die Sekunde genau piepst. Auf diese Weise „stiehlt" der Wecker nur wenig Rechenzeit. □

```
cseg        segment 'code'
            assume  cs:cseg,ss:sseg

Marker      db      'WECKER'
Minutes     db      0
Hours       db      0
Active      db      0           ; 0: arbeitsbereit
Counter     db      0
ContPtr     dd      0           ; Adresse der alten INT 1Ch Routine
HM          db      0           ; 0: Stundeneingabe, 1: Minuteneingabe
Beep        db      0           ; 0: Kein Pieps, ansonsten läuft Pieps
Beep09      db      0           ; Zähler für 9 Interrupts (= 1/2 Sekunde)

Work        proc
            push    ax
            push    cx
            push    dx
            cmp     Beep,0
            jne     Bell
            inc     Counter
            jne     WorkQ
            cmp     Active,0
            jne     WorkQ
            cmp     word ptr Minutes,0
            je      WorkQ

            mov     Active,1    ; in Arbeit
            sti                 ; Interrupts erlauben
            mov     ah,0        ; Zeit nach CX:DX holen
            int     1Ah
            mov     ax,dx       ; Zeit nach DX:AX übertragen
            mov     dx,cx
            mov     cx,1092     ; 1092 = 18.2 * 60
            div     cx
            mov     cl,60       ; AX enthält Zeit in Minuten
            div     cl          ; Umrechnung in Stunden und Minuten
            xchg    al,ah       ; AH: Stunden, AL: Minuten
            cmp     ax,word ptr Minutes ; Zeit vergleichen
            jb      WorkQ
            mov     word ptr Minutes,0   ; Wecker ausschalten
            mov     Beep,11              ; Piepsen aktivieren
Bell:
            cmp     Beep09,0
            ja      DecBell      ; Dekrementieren bis 1/2 Sek. vergangen
            mov     Beep09,9
            dec     Beep
            test    Beep,1
            je      SpkOff
SpkOn:
            cli
            mov     al,0B6h      ; Frequenz setzen
            out     43h,al
            mov     al,5
            out     42h,al
            out     42h,al
            in      al,61h
            or      al,00000011b ; Lautsprecher einschalten
            out     61h,al
            sti
            jmp     short WorkQ
SpkOff:
            cli
            in      al,61h
            and     al,11111100b ; Lautsprecher ausschalten
            out     61h,al
            sti
            jmp     short WorkQ
DecBell:
            dec     Beep09
WorkQ:      pop     dx
            pop     cx
```

Bild 1. Mit diesem Programm wird Pünktlichkeit vom Rechner unterstützt

```
                pop     ax
                cli                     ; Interrupts verbieten
                mov     Active,0        ; nicht in Arbeit
                jmp     cs:ContPtr      ; alte INT 1Ch Routine aufrufen
Work            endp

Init            proc                    ; Initialisierungsteil
                push    ds              ; Segment des PSP retten
                mov     ax,cseg         ; DS laden
                mov     ds,ax
                mov     ax,900h                 ; Meldung ausgeben
                mov     dx,offset NormMsg
                int     21h
                pop     ds              ; Segment des PSP wiederherstellen
                cld
                mov     si,80h
                lodsb                   ; Anzahl der Zeichen holen
                mov     ah,0            ; AH : Aktuelle Rechenvariable
                mov     cx,ax
                add     cx,80h
                mov     dx,0            ; DH: Stunden, DL: Minuten
Skip:
                cmp     si,cx           ; Nächstes Zeichen holen
                ja      EndPar
                lodsb                   ; Zeichen nach AL
                cmp     al,' '          ; wenn Leerzeichen, nächstes Zeichen holen
                je      Skip
                cmp     al,':'          ; Stunden/Minuten Begrenzer
                je      ChHM
                cmp     al,'.'          ; Stunden/Minuten Begrenzer
                je      ChHM
                cmp     al,'0'          ; wenn Zeichen kleiner 0, Fehler
                jb      ParErr
                cmp     al,'9'          ; wenn Zeichen größer 9, Fehler
                ja      ParErr
                inc     Counter         ; wenn mehr als 2 Zeichen, Fehler
                cmp     Counter,2
                ja      ParErr
                sub     al,'0'
                mov     bh,al
                mov     al,ah
                mov     bl,10           ; Bisherigen Stand mit 10 multiplizieren
                mul     bl
                add     al,bh           ; und Zeichen hinzuaddieren
                mov     ah,al
                jmp     short Skip ; Nächstes Zeichen
ParErr:
                mov     ax,cseg         ; Fehler Routine
                mov     ds,ax
                mov     ax,900h
                mov     dx,offset ErrMsg
                int     21h
                jmp     InitQ
ChHM:                                   ; Stunden Zwischenspeicherung
                cmp     HM,0            ; wenn bereits Stunden gespeichert, Fehler
                jne     ParErr
                mov     Counter,0       ; Zeichenzähler auf 0 setzen
                mov     HM,1            ; Minuteneingabe markieren
                mov     dh,ah           ; Stunden sichern
                mov     ah,0            ; Minuten löschen
                jmp     short Skip
EndPar:                                 ; Ende der Parameterabfrage
                cmp     HM,0
                je      EndPar2
                mov     dl,ah           ; Minuten speichern
                jmp     short EndPar3
EndPar2:
                mov     dh,ah           ; Stunden speichern
EndPar3:
                mov     word ptr Minutes,dx
                push    ds                      ; PSP Adresse retten
                mov     ax,cseg
                mov     ds,ax
                mov     ax,351Ch        ; Prüfen, ob bereits installiert
                int     21h
                mov     word ptr ContPtr,bx     ; Adr. der alten Routine speichern
                mov     word ptr ContPtr+2,es
                mov     di,offset Marker        ; Erkennungs-String vergleichen
                mov     si,di
                mov     cx,6
                repe    cmpsb
                je      InitNR

                mov     ax,251Ch        ; Residente Installierung
                mov     dx,offset Work
                int     21h
                mov     ax,900h
                mov     dx,offset InstMsg
                int     21h
                pop     ax              ; Anzahl der Paragraphen berechnen
                mov     dx,offset Init
                mov     cl,4
                shr     dx,cl
                inc     dx
                mov     cx,cs
                add     dx,cx
                sub     dx,ax
                mov     ax,3103h        ; Resident beenden
                int     21h
InitNR:
                pop     ax                      ; Nicht residente Beendung
                mov     dx,word ptr Minutes ; Zeit in residente Routine übertragen
                mov     es:word ptr Minutes,dx
InitQ:
                mov     ax,4C00h        ; Nicht-Resident beenden
                int     21h
Init            endp

NormMsg         db      0Dh,0Ah
                db      '    Wecker Version 2.01',0Dh,0Ah
                db      '    Syntax: WECKER SS:MM',0Dh,0Ah
                db      '            WECKER schaltet aus!',0Dh,0Ah,'$'
InstMsg         db      0Dh,0Ah
                db      '    Residenter Teil des Weckers installiert!',0Dh,0Ah,'$'
ErrMsg          db      0Dh,0Ah
                db      '    FEHLER: Zeitangabe ungültig!',0Dh,0Ah
                db      '    Weckereinstellung nicht geändert!',0Dh,0Ah,'$'

cseg            ends

sseg            segment stack 'stack'
                db      20 dup ('Alarm-stack')
sseg            ends

                end     Init
```

```
procedure Wecker (SS,MM : integer);

var
   Test : string [6];
   i    : integer;
   Reg  : record
             AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : integer;
          end;
begin
   Reg.AX := $351C; (Segment-Adresse ermitteln)
   MsDos (Reg);
   Test := '';
   for i := 0 to 5 do
      Test := Test + chr (Mem [Reg.ES:i]);
   if Test = 'WECKER' then
   begin
      Mem [Reg.ES:7] := SS;
      Mem [Reg.ES:6] := MM;
   end;
end;
```

Bild 2. Auch von Turbo-Pascal aus läßt sich der Wecker benutzen

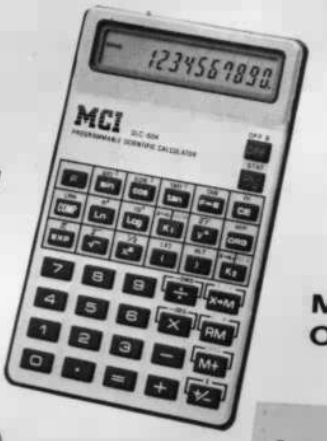
... KOMPLETTPREISE...SYSTEMPAKETE

KOMPLETTPREISE...SYSTEMPAKETE...1 JAHR GARANTIE

System Pakete für kluge Rechner

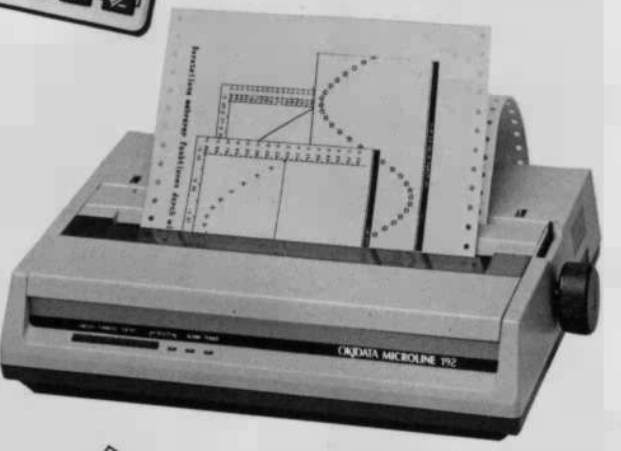
ab 1349,–

+



(+)

MCI Printer Plus **350,–**
OKI ML192 Elite **899,–**



System Paket 11

10 MHz

1349,–

- **MCI XT16SLC,** 640 K, 1 x 360 K, Clock, ser. par., 12" Monitor, MS-DOS, Tastatur
- **MCI** Programmierbarer Taschenrechner

opt. MCI 120 Zeichen Printer oder OKI ML 192 Elite (Mehrpreis siehe oben)

System Paket 21

10 MHz

1899,–

- **MCI XT16SLC,** 640 K, 1 x 360 K, Clock, ser. par., 12" Monitor, MS-DOS, Tastatur
- **MCI** 20MB Festplatte m. System formatiert
- **MCI** Programmierbarer Taschenrechner

opt. MCI 120 Zeichen Printer oder OKI ML 192 Elite (Mehrpreis siehe oben)

System Paket 31

12 MHz

2399,–

- **MCI AT4SLC,** 640 K, 1 x 1, 2 MB, Clock, ser. par., 12" Monitor, MS-DOS, Tastatur
- **MCI** Programmierbarer Taschenrechner

opt. MCI 120 Zeichen Printer oder OKI ML 192 Elite (Mehrpreis siehe oben)

System Paket 41

12 MHz

3199,–

- **MCI AT4SLC,** 640 K, 1 x 1, 2 MB, Clock ser. par., 12" Monitor, MS-DOS, Tastatur
- **MCI** 20MB Festplatte m. System formatiert
- **MCI** Programmierbarer Taschenrechner

opt. MCI 120 Zeichen Printer oder OKI ML 192 Elite (Mehrpreis siehe oben)

PREIS... QUALITÄT... 1 JAHR GARANTIE

KOMPATIBEL ... 24-STUNDEN-TEST ... LEISTUNG ...

MCI XT 16 SLC

alles drin!

Grundausstattung ohne Monitor

ab **899,–**

beinhaltet:

- voll IBM®XT kompatibel
- 8088 CPU + 8087 Sockel
- 8 XT Slots
- 256 KB freier Speicher
- 1 x 360 KB Floppy-Drive
- Color- oder Monochr. Grafikkarte (Hercules II komp. 720 x 348 P.)
- Deutsche Normtastatur MK 5111
- 150 W Schaltnetzteil
- Parallele Drucker-Schnittstelle

Erweiterungen für XT 16 SLC-Serie

2. Laufwerk 360 KB	249,–
Speichererweiterung auf 640 KByte	149,–
Clock/Seriell-Karte	79,–
I/0 Plus II Karte	149,–
20 MB Festplatte mit XT-Controller	+ 589,–
30 MB Festplatte m. RLL XT-Contr.	+ 679,–
EGA-Set statt monochr. Karte	+ 1149,–
Opt. Roll-Maus MO 86 m. Softw.	+ 199,–
Professional Multifunktions-Tastatur MK 6000	+ 100,–
MS-DOS 3.2 + GW-Basic	+ 149,–
9" TTL Monitor grün	+ 150,–
12" Monitor grün od. bern.	+ 229,–
14" TTL Monitor grün, bern. od. weiß	+ 279,–

MCI AT 4 SLC

alles drin!

Grundausstattung ohne Monitor

ab **1899,–**

beinhaltet:

- voll IBM® AT kompatibel
- 80286 CPU + 80287 Sockel
- 6 AT + 2 XT Slots
- 8 und **12 MHz** umschaltbar
- 512 KB freier Speicher
- 1 x 1,2 MB/360 KB Laufwerk
- Color- oder Monochr. Grafikkarte (Hercules II komp. 720 x 348 P.)
- Parallele Drucker-Schnittstelle
- Batteriegep. Echtzeituhr/Kalender
- Kapazitive deutsche Normtastatur

Erweiterungen für AT 4 SLC-Serie

2. Laufwerk 360 KB	299,–
20 MB Festplatte mit AT-Controller	889,–
30 MB Festplatte m. RLL AT-Contr.	+ 999,–
Seriell-Karte	79,–
I/0 Plus II Karte	149,–
EGA-Set statt monochr. Karte	+ 1149,–
MS-DOS 3.2 + GW-Basic	+ 149,–
Professional Multifunktions-Tastatur MK 6000	+ 100,–
9" TTL Monitor grün	+ 150,–
12" Monitor grün od. bern.	+ 229,–
14" TTL Monitor grün, bern./weiß	+ 279,–

PRINTER

MCI *Personal Computer Graphics Printer Plus*

- **voll kompatibel zum** *IBM Personal Computer Graphics Printer*
- **120 Zeichen/sec.**

399,–

OKI MICROLINE ML 192 Elite

- 9 Nadel Matrixdrucker
- Druckgeschwindigkeit 200 Z./sec.
- 40 Zeichen/sec. NLQ
- Druckpuffer 8 KB
- IBM Kompatibel

899,–

EGA

Hochauflösender EGA-Monitor

EGA-Karte 249,–

- Auflösung
  320 x 200 (CGA Mode)
  640 x 350 (EGA Mode)

999,–

TELEFON-HOTLINE-PREISE
MCI Telefonansage (20 sec.)

	(0 22 02)
Festplatten & Controller	10 81 40
Grafikkarten	10 81 41
Monitore	10 81 42
Schnittstellenkarten	10 81 43
Drucker	10 81 44
Aktuelle Neuigkeiten	10 81 45

MCI

5060 Bergisch Gladbach 2
Bensberger Straße 252
Tel.-Nr.: 02202/1080
Fax: 02202/31009 · Telex: 8873518

12 Monate Garantie auf alle Geräte. Nach der Pang Vo. v. 14. 3. 85 sind wir bei Angeboten gegenüber dem Endverbraucher zur Angabe der **Preise incl. MwSt.** verpflichtet. Für Druckfehler wird nicht gehaftet. **Preise gültig ab 1. 1. 88.** Lieferzeit und Lieferbedingungen auf Anfrage. Änderungen, die technischen Verbesserungen dienen, vorbehalten. Zwischenverkauf vorbehalten.
MCI MICRO COMPUTER INSTRUMENTS GMBH eingetragen AG Bergisch Gladbach - HRB 2575. Herstellung und Vertrieb von Mikrocomputern. 5060 Bergisch Gladbach 2 - Bensberger Straße 252

Hoen-oh Shin, Heiko Stuckenberg, Tobias Weltner

Zeichnen von Karikaturen

Ein Basic-Programm für den Amiga

Im folgenden Beitrag zeigen die Autoren anhand eines Programmes Techniken, mit denen man auch ohne große künstlerische Fähigkeiten Karikaturen von lieben Mitmenschen anfertigen kann. Mit einem Amiga und zwei interessanten Routinen geht das recht fix vonstatten.

Zuerst stellt sich die Frage, was einen guten Karikaturisten auszeichnet. Nun, zweifelsohne sein Gefühl für charakteristische Merkmale einer Person, die der Zeichner dann hervorhebt und überbetont, um ein satirisches Zerrbild des Originals zu fabrizieren. Das Hauptproblem bei der Programmierung ist also, dem Computer beizubringen, wie er charakteristische Merkmale eines Gesichts findet und sie übertrieben darstellt. Was Karikaturisten mit viel Menschenkenntnis und Berufserfahrung erkennen, müssen wir simulieren, indem wir versuchen, diese scheinbar irrationalen Eigenschaften zu formalisieren.

```
AnzeigeX = GesichtX + faktor * (GesichtX - NormX)
AnzeigeY = GesichtY + faktor * (GesichtY - NormY)
```

Bild 1. Das Prinzip, nach dem der Rechner die Karikatur errechnet: Die Abweichung von der Norm (rechts in Klammern) wird, mit einem Faktor multipliziert, zum ursprünglichen Bild addiert

```
AnzeigeX = GesichtX - faktor * (GesichtX - NormX)
AnzeigeY = GesichtY - faktor * (GesichtY - NormY)
```

Bild 2. Durch Verringerung der Unterschiede lassen sich ebenso zwei Gesichter mischen

Ein Schema

Dazu bietet sich folgender Ansatz an: Charakteristische Kennzeichen eines Gesichts sind einzigartige Merkmale einer Person, die sie von den anderen unterscheidet – mit anderen Worten, Abweichungen von der *Norm*. Wenn wir nun unseren heißgeliebten Blech- bzw. Plastikkasten mit einer Normgröße füttern und verschiedene Gesichter damit vergleichen lassen, findet der Computer die Abweichungen heraus und übertreibt sie. Auf diesem Lösungsansatz basiert das Programm.
Ein Normgesicht und ein Vergleichsgesicht, die jeweils aus 186 Punkten bestehen (ich weiß, ich weiß, aber die Tipparbeit lohnt sich wirklich!), werden an jedem Punkt miteinander verglichen und die Abweichung (Differenz der Koordinaten) mit einem „Karikierfaktor" multipliziert. Anschließend wird das Ergebnis addiert. Dieser Punkt stellt dann eine Koordinate der Karikatur dar *(Bild 1)*.

In einem mathematischen Modell kann man sich die Funktionsweise dieser Routine folgendermaßen vorstellen: Jedes Gesicht stellt einen Punkt in einem 372-dimensionalen Raum dar (372-dimensional, weil jeder Stützpunkt aus einer X- und Y-Koordinate besteht; akribisch veranlagten Lesern steht es natürlich frei, detailliertere Bilder anzufertigen und in höheren Sphären – sprich Dimensionen – zu schweben). In diesem Raum erkennt man ähnliche Gesichter daran, daß sie dicht beieinander liegen (z. B. Zwillinge). Man kann also den Betrag der Länge einer Verbindungslinie zwischen zwei Punkten als Gradmesser für die Ähnlichkeit zweier Gesichter ansehen. Die Routine tut nichts anderes als den Punkt, der das Gesicht repräsentiert, an der Verbindungslinie entlang zu verschieben – und zwar weiter weg vom Normgesicht, wodurch die Karikatur entsteht (der Punkt bleibt bei der Verschiebung auf der Linie, weil die Koordinaten proportional zueinander verändert werden).

Vielleicht haben Sie schon die zweite interessante Anwendung der Routine entdeckt: Man kann die Entfernung zwischen zwei Punkten nicht nur vergrößern – bei einer Verkürzung, also einer Verschiebung des Vergleichsgesichts näher zum Normgesicht, entsteht ein Mischgesicht, das Elemente beider Gesichter enthält. Für diese Anwendung muß man die Karikatur-Routine nur leicht verändern *(Bild 2)*. Natürlich ist auch eine Vermischung mehrerer Gesichter möglich, doch muß man dann andere, etwas kompliziertere Routinen verwenden. Im übrigen ist man nicht allein auf menschliche Gesichter beschränkt: Interessante Bilder ergeben sich auch durch Mischen von Affengesichtern mit menschlichen oder einer Birne mit... (Sie wissen schon). Eine andere reizvolle Perspektive ergibt sich in unserem Karikaturprogramm, wenn man das Normgesicht wechselt. Dadurch findet ein Standpunktwechsel statt, und die Karikaturen erhalten eine andere Note. Wenn man z. B. als Normgesicht einen Schimpansen nimmt, entstehen Karikaturen mit Schlitzaugen, spitzen Nasen und wulstigen Lippen, weil diese Punkte im Vergleich zum normalen Affengesicht besonders hervortreten.

Das Programm

Das vorliegende Programm *(Bild 3)* dürfte nicht allzuschwer zu verstehen sein, die beiden wichtigsten Prozeduren (KARIKATUR u. MISCHEN) wurden ja bereits erläutert. Der Programmablauf gestaltet sich folgendermaßen: Zuerst wird durch die Prozeduren INIT-SCREEN und INIT-VARIABLES das Programm initialisiert. Danach erscheint das Hauptmenü mit den acht Unterprogrammen. Zu beachten ist, daß immer zwei Gesichter eingeladen oder eingegeben sein müssen, bevor die Prozeduren KARIKATUR oder MISCHEN aufgerufen werden können. Das Normgesicht muß dabei in den Speicher 1 geladen werden.

Mit DIGITIZER lassen sich neue Gesichter eingeben. Als Eingabemedium wird hier die Maus verwendet, mit der man nur die Referenzpunkte anklickt. Damit entfallen umständliche numerische Eingaben. Allerdings muß sehr sorgfältig auf die Reihenfolge der Eingabe geachtet werden. Aus Platzgründen können wir leider keine ausführliche Angaben über die Reihenfolge der Eingabe machen. Es ist aber einfach, die Folge festzustellen, indem man das Normgesicht im TRACE-Mode zeichnen läßt und auf ei-

```
'* exec.-Funktionen werden nur fuer Hardcopy gebraucht
DECLARE FUNCTION AllocMem& LIBRARY
DECLARE FUNCTION DoIO% LIBRARY
DECLARE FUNCTION OpenDevice% LIBRARY
DECLARE FUNCTION AllocSignal% LIBRARY
DECLARE FUNCTION FindTask& LIBRARY

LIBRARY "exec.library"

'****  Begruessung,Fenster oeffnen,etc. ****

GOSUB INIT.SCREEN

'****  Felder dimensionieren fuer ****

'* Namen der Gesichtsteile (Auge,Mund,etc.)
DIM merkmale$(39)
'* und wieviel Punkte das Merkmal hat
DIM merkmale(39)
'* Hier stehen die Daten
'* 1. Dimension  Gesicht1/Gesicht2
'* 2. Dimension  Alle Punkte eines Gesichts
'* 3. Dimension  X/Y - Koordinate
DIM gesicht(2,186,2)
'* Hier kommt ein berechnetes Bild rein
DIM anzeige(186,2)
'* und hier stehen die Namen der Gesichter,
'die auch beim Laden und Speichern benutzt werden
DIM gesicht$(2)

'* Datas lesen, schon mal Normgesicht laden

GOSUB INIT.VARIABLES

'****  Hauptmenue  ****

wahl$=""
WHILE (wahl$<>"8")
  CLS
  PRINT "Hauptmenue:"
  PRINT
  PRINT "1. Digitizer"
  PRINT "2. Gesicht laden"
  PRINT "3. Gesicht sichern"
  PRINT "4. Karikatur erstellen"
  PRINT "5. Gesichter mischen"
  PRINT "6. Bild anzeigen"
  PRINT "7. Gesicht anpassen"
  PRINT "8. Ende"
  PRINT
  PRINT "Geladene Gesichter :"
  PRINT "Speicher 1  -  ";gesicht$(1)
  PRINT "Speicher 2  -  ";gesicht$(2)
  PRINT
  INPUT "Was soll's sein ";wahl$
  IF wahl$>"0" OR wahl$<"9" THEN
    wahl=VAL(wahl$)
    ON wahl GOSUB DIGITIZER,Laden,SAVEN,Karikatur,MISCHEN,AUSGABE,ANPASSEN
  END IF
WEND
END
```

Bild 3. Das Programm „Karakiri" – so strukturiert kann man in Amiga-Basic programmieren

```
INIT.SCREEN:
 '* Das Fenster nach dem Oeffnen manuell auf volle Groesse ausdehnen !
  WINDOW 1,"Karakiri",(0,0)-(500,150),1,-1
  CLS
  PALETTE 0,1,1,1
  PALETTE 1,0,0,0
  PRINT
  PRINT SPACE$(27);"Karakiri"
  PRINT
  PRINT "  ";CHR$(169);
  PRINT " 1987 by Heiko Stuckenberg & Hoen-oh Shin & Tobias Weltner"
  '* Faktoren, um rechteckige Pixels an Amigagrafik anzupassen
  '* und das Bild an die rechte Stelle zu ruecken
  xfak=1.7
  yfak=.8
  xsch=-40
  ysch=-10
RETURN

INIT.VARIABLES:
  RESTORE
  FOR i=1 TO 39
    READ merkmale$(i),merkmale(i)
  NEXT i
  filename$="Normgesicht.kar"
  gesicht$(1)="Normgesicht"
  gesicht$(2)=""
  gesichtsauswahl=1
  GOSUB LADE.GESICHT

  DATA "Linke Pupille",1,"Rechte Pupille",1,"Linke Iris",5,"Rechte Iris",5
  DATA "Unterer Rand linkes Augenlid",3,"Unterer Rand rechtes Augenlid",3
  DATA "Unterer Rand linkes Auge",3,"Unterer Rand rechtes Auge",3
  DATA "Oberer Rand linkes Auge",3,"Oberer Rand rechtes Auge",3
  DATA "Linke Augenlinie",3,"Rechte Augenlinie",3
  DATA "Linke Seite der Nase",6,"Rechte Seite der Nase",6
  DATA "Linkes Nasenloch",6,"Rechtes Nasenloch",6
  DATA "Oberer Rand linke Augenbraue",6,"Oberer Rand rechte Augenbraue",6
  DATA "Unterer Rand linke Augenbraue",4,"Unterer Rand rechte Augenbraue",4
  DATA "Oberer Rand der Oberlippe",7,"Unterer Rand der Oberlippe",7
  DATA "Oberer Rand der Unterlippe",7,"Unterer Rand der Unterlippe",7
  DATA "Linke Gesichtsseite",3,"Rechte Gesichtsseite",3
  DATA "Linkes Ohr",7,"Rechtes Ohr",7
  DATA "Kiefer",11,"Haarlinie",13,"Obere Begrenzung des Gesichts",13
  DATA "Linke Wangenlinie",3,"Rechte Wangenlinie",3
  DATA "Linker Backenknochen",3,"Rechter Backenknochen",3
  DATA "Linke obere Lippenlinie",2,"Rechte obere Lippenlinie",2
  DATA "Kinnspalte",2,"Kinnlinie",3
RETURN

'****  Laden und Speichern wurde in zwei Teile geteilt,
 damit das Normgesicht zu Beginn gleich eingeladen werden kann ****

Laden:
  INPUT "Welches Gesicht laden (Name) ";eingabe$
  IF eingabe$="" THEN RETURN
  INPUT "Und in welchen Speicher laden (1/2) ";gesichtsauswahl
  gesicht$(gesichtsauswahl)=eingabe$
  filename$=eingabe$+".kar"
  GOSUB LADE.GESICHT
RETURN

SAVEN:
  INPUT "Welches Gesicht sichern (1/2) ";eingabe$
  IF eingabe$="" THEN RETURN
  gesichtsauswahl=VAL(eingabe$)
  filename$=gesicht$(gesichtsauswahl)+".kar"
  GOSUB SAVE.GESICHT
RETURN
```

```
Karikatur:
  '* Eins als Faktor ist schon viel !!!
  INPUT "Karikaturfaktor ";eingabe$
  IF eingabe$="" THEN RETURN
  faktor=VAL(eingabe$)
  '* Gesicht 1 = Normgesicht ; Gesicht 2 = zu karikierendes Gesicht
  FOR punkt=1 TO 186
    anzeige(punkt,1)=gesicht(2,punkt,1)+faktor*(gesicht(2,punkt,1)-gesicht(1
,punkt,1))
    anzeige(punkt,2)=gesicht(2,punkt,2)+faktor*(gesicht(2,punkt,2)-gesicht(1
,punkt,2))
  NEXT punkt
RETURN

AUSGABE:
  '* Ausgegeben wird das Array Anzeige,
  '* in dem die Karikatur oder ein Mischgesicht steht
  '* Hier wird jetzt auch die Anpassung an die Amigagrafik beruecksichtigt
  CLS
  '* die Pupillen werden als Kreise ausgefuehrt,
  '* und koennten auch ausgefuellt werden
  CIRCLE (xfak*135+xsch,yfak*145+ysch),2
  CIRCLE (xfak*190+xsch,yfak*145+ysch),2
  '* einzelne Merkmale des Gesichts durchgehen,
  '* und die Punkte eines Merkmals miteinander durch Linien verbinden
  punkt=2
  FOR merkmal=3 TO 39
    punkt=punkt+1
    position=1
    WHILE position<merkmale(merkmal)
      LINE (xfak*anzeige(punkt,1)+xsch,yfak*anzeige(punkt,2)+ysch) - (xfak*a
nzeige(punkt+1,1)+xs
ch,yfak*anzeige(punkt+1,2)+ysch)
      position=position+1
      punkt=punkt+1
    WEND
  NEXT merkmal
  LOCATE 1,1:INPUT "Bild ausdrucken (j/n) ";jn$
  IF jn$="j" OR jn$="J" THEN Ausdrucken
RETURN

DIGITIZER:
  CLS
  '* Welchen Speicher verwenden (1 oder 2) ?
INPUT "Welches Gesicht (1/2) ";eingabe$
IF eingabe$="" THEN RETURN
gesicht%=VAL(eingabe$)
CLS
LOCATE 1,1:PRINT "OK. Geben Sie nun die Referenzpunkte an!"+SPACE$(30)
in%=0
'* Merkmale per Mauskoordinaten einlesen
'* nach Merkmalen geordnet (loop% geht die Merkmalsliste durch)
FOR loop%=1 TO 39
  LOCATE 3,1:PRINT "Merkmal: ";merkmale$(loop%)+SPACE$(40)
  '* zaehler = Anzahl der Punkte fuer dieses Merkmal
  zaehler%=merkmale(loop%)
  '* zaehler% wird dekrementiert, wenn 0 dann naechstes Merkmal
  WHILE zaehler%>0
    LOCATE 4,1:PRINT "Referenzpunkt: ";zaehler%;"   "
    GetMaus in1%,in2%,in$
    '* Wenn Taste '-' gedrueckt wurde,
    '* dann diesen Referenzpunkt nochmal eingeben
    IF in$="-" THEN
      zaehler%=zaehler%+1
      in%=in%-1
    END IF
    '* in% = Punktezaehler (1-186)
    in%=in%+1
    '* Referenzpunkt im Bildschirmformat abspeichern
    gesicht(gesicht%,in%,1)=in1%
    gesicht(gesicht%,in%,2)=in2%
    zaehler%=zaehler%-1
    '* wenn schon ein Punkt des Merkmales existiert, dann Linie ziehen...
    IF merkmale(loop%)>1 AND zaehler%<merkmale(loop%)-1 THEN
      LINE(gesicht(gesicht%,in%,1),gesicht(gesicht%,in%,2))-(gesicht(gesic
ht%,in%-1,1),gesicht
(gesicht%,in%-1,2))
    '* ...sonst nur Punkt zeichnen
    ELSE
      PSET(gesicht(gesicht%,in%,1),gesicht(gesicht%,in%,2))
    END IF
  WEND
 NEXT loop%
 GOSUB ANPASSEN
RETURN

MISCHEN:
  INPUT "Wieviel Prozent von Gesicht 1 ";eingabe$
  IF eingabe$="" THEN RETURN
  proz=VAL(eingabe$)
  '* Wir bilden uns hier einen Vektor von Gesicht 1 nach Gesicht 2,
  '* auf dem wir entlanggehen werden.
  '* Dazu muessen wir die Prozentzahl von 100 Prozent abziehen,
  '* obwohl man natuerlich auch in anderer Richtung auf dem Vektor
  '* entlanggehen koennte
  faktor=1-proz/100
  FOR punkt=1 TO 186
    '* Es ist natuerlich fuer jeden Gesichtspunkt ein neuer Vektor
    xvektor=gesicht(2,punkt,1)-gesicht(1,punkt,1)
    yvektor=gesicht(2,punkt,2)-gesicht(1,punkt,2)
    anzeige(punkt,1)=gesicht(1,punkt,1)+xvektor*faktor
    anzeige(punkt,2)=gesicht(1,punkt,2)+yvektor*faktor
  NEXT punkt
RETURN

'**** Eigentliche Laderoutine ****
'* die auch direkt von INIT.VARIABLES angesprungen wird

LADE.GESICHT:
  OPEN filename$ FOR INPUT AS 4
  punkt=1
  FOR merkmal=1 TO 39
    FOR j=1 TO merkmale(merkmal)
      INPUT #4,gesicht(gesichtsauswahl,punkt,1)
      INPUT #4,gesicht(gesichtsauswahl,punkt,2)
      punkt=punkt+1
    NEXT j
  NEXT merkmal
  CLOSE #4
RETURN

SAVE.GESICHT:
  OPEN filename$ FOR OUTPUT AS 4
  punkt=1
  FOR merkmal=1 TO 39
    FOR j=1 TO merkmale(merkmal)
      PRINT #4,gesicht(gesichtsauswahl,punkt,1)
      PRINT #4,gesicht(gesichtsauswahl,punkt,2)
      punkt=punkt+1
    NEXT j
  NEXT merkmal
  CLOSE #4
RETURN
```

```
ANPASSEN:
  '* setzt ein Bild mit beliebigem Augenabstand
  "* und beliebiger Verschiebung der Augenkoordinaten so um,
  '* dass die Koordinaten fuer das Programm nutzbar werden,
  '* da zumindest die Augen an den gleichen Koordinaten liegen muessen
  INPUT "Welchen Speicher anpassen (1/2) ";eingabe$
  IF eingabe$="" THEN RETURN
  gesichtsauswahl=VAL(eingabe$)
  '* Verschiebung um linkes Auge auf Position zu bringen
  xverschieb=135-gesicht(gesichtsauswahl,1,1)
  yverschieb=145-gesicht(gesichtsauswahl,1,2)
  '* und erstmal das ganze Gesicht verschieben
  FOR punkt=1 TO 186
    gesicht(gesichtsauswahl,punkt,1)=gesicht(gesichtsauswahl,punkt,1)+xversc
hieb
    gesicht(gesichtsauswahl,punkt,2)=gesicht(gesichtsauswahl,punkt,2)+yversc
hieb
  NEXT punkt
  '* jetzt einen Streckfaktor bestimmen,
  '* so dass das rechte Auge an seinen Platz kommt
  '* diese Streckung/Stauchung muss jetzt auf jeden Punkt angewendet werden,
  '* damit die Proportionen stimmen
  streck=55/(gesicht(gesichtsauswahl,2,1)-gesicht(gesichtsauswahl,1,1))
  FOR punkt=2 TO 186
    gesicht(gesichtsauswahl,punkt,1)=135+INT((gesicht(gesichtsauswahl,punkt,
1)-135)*streck+.5)
    gesicht(gesichtsauswahl,punkt,2)=145+INT((gesicht(gesichtsauswahl,punkt,
2)-145)*streck+.5)
  NEXT punkt
RETURN

SUB GetMaus(a%,b%,kor$) STATIC
     '* wartet, bis linke Maustaste gedrueckt UND wieder losgelassen
     '* wurde und liefert dann die x- und y-Koordinaten der Maus sowie
     '* evtl. gedrueckte Taste
      mc% = MOUSE(0)
      kor$=""
      WHILE mc% < 1
        mc% = MOUSE(0)
        IF kor$="" THEN kor$=INKEY$
      WEND
      a% = MOUSE(1)
      b% = MOUSE(2)
END SUB

SUB Ausdrucken STATIC
     '* druckt den Inhalt des BASIC-Fensters als Hardcopy aus
     '* Speicher besorgen
      o& = 2^0+2^16
      io& = AllocMem&(100,o&)
      port&  = io&+62
      IF io& = 0 THEN ERROR 7

     '* Fensterparameter auslesen
      fen&   = WINDOW(7)
      ras&   = PEEKL(fen&+50)
      br%    = PEEKW(fen&+112)
      ho%    = PEEKW(fen&+114)
      scr&   = PEEKL(fen&+46)
      vp&    = scr&+44
      colmap&  = PEEKL(vp&+4)
      vpm%     = PEEKW(vp&+32)


      sB% = AllocSignal%(-1)
      IF sB% = -1 THEN ERROR 255
      sTask& = FindTask&(0)

     * Port einrichten
      POKEL port&+10,port&+34
      POKE  port&+15,sB%
      POKEL port&+16,sTask&
      POKEL port&+20,port&+24
      POKEL port&+28,port&+20
      POKE  port&+34,ASC("k")
      CALL  AddPort(port&)

     * IODRP-Struktur
      POKE  io&+8,5
      POKEL io&+14,port&
      POKEW io&+28,11
      POKEL io&+32,ras&
      POKEL io&+36,colmap&
      POKEL io&+40,vpm%
      POKEW io&+48,br%
      POKEW io&+50,ho%
      POKEL io&+52,br%
      POKEL io&+56,ho%

     '* Drucker oeffnen
      dev$ = "printer.device"+CHR$(0)
      s% = OpenDevice%(SADD(dev$),0,io&,0)
      IF s%<>0 THEN
        PRINT "Drucker noch offen."
        CALL  FreeMem(io&,100)
        CALL  FreeSignal(sB%)
        EXIT SUB
      END IF

     * Ausdrucken und Resourcen zurueckgeben
      f% = DoIO%(io&)
      CALL CloseDevice(io&)
      CALL RemPort(port&)
      CALL FreeMem(io&,100)
      CALL FreeSignal(sB%)
      PRINT "Fehlercode: ";f%
END SUB
'****  E N D E   (' Geschafft !?!?! ')  ****
```

```
'**** Hilfsprogramm zur Eingabe des Normgesichts ****

OPEN "Normgesicht.kar" FOR OUTPUT AS 4
FOR i=1 TO 186
  PRINT "Punkt ";i;" , X-Koordinate ";
  INPUT a
  PRINT #4,a
  INPUT "Y-Koordinate ";a
  PRINT #4,a
NEXT i
CLOSE #4
```

Bild 4. Mit diesem Hilfsprogramm werden die Werte aus Bild 5 und 6 eingegeben

```
( 135 , 145 ),( 190 , 145 ),( 134 , 141 ),( 128 , 144 ),( 133 , 149 ),
( 140 , 144 ),( 135 , 141 ),( 190 , 141 ),( 184 , 144 ),( 189 , 149 ),
( 196 , 144 ),( 190 , 141 ),( 119 , 147 ),( 133 , 140 ),( 147 , 146 ),
( 177 , 147 ),( 190 , 141 ),( 203 , 147 ),( 121 , 147 ),( 133 , 150 ),
( 147 , 146 ),( 177 , 147 ),( 191 , 150 ),( 201 , 148 ),( 118 , 143 ),
( 132 , 137 ),( 148 , 142 ),( 176 , 143 ),( 191 , 137 ),( 204 , 143 ),
( 127 , 154 ),( 135 , 153 ),( 144 , 150 ),( 178 , 151 ),( 187 , 154 ),
( 196 , 154 ),( 156 , 140 ),( 156 , 153 ),( 156 , 165 ),( 154 , 172 ),
( 156 , 179 ),( 161 , 182 ),( 166 , 140 ),( 166 , 153 ),( 166 , 166 ),
( 168 , 172 ),( 167 , 179 ),( 161 , 182 ),( 150 , 169 ),( 147 , 173 ),
( 146 , 178 ),( 148 , 182 ),( 153 , 179 ),( 161 , 182 ),( 173 , 169 ),
( 176 , 172 ),( 177 , 178 ),( 174 , 182 ),( 170 , 179 ),( 163 , 182 ),
( 112 , 137 ),( 113 , 132 ),( 125 , 127 ),( 139 , 128 ),( 150 , 131 ),
( 152 , 136 ),( 171 , 136 ),( 173 , 132 ),( 186 , 129 ),( 199 , 128 ),
( 208 , 132 ),( 211 , 137 ),( 112 , 138 ),( 124 , 132 ),( 138 , 134 ),
( 152 , 136 ),( 171 , 136 ),( 187 , 134 ),( 200 , 132 ),( 210 , 137 ),
( 137 , 203 ),( 149 , 199 ),( 156 , 196 ),( 162 , 199 ),( 168 , 197 ),
( 177 , 199 ),( 187 , 202 ),( 138 , 203 ),( 148 , 203 ),( 156 , 202 ),
( 163 , 203 ),( 170 , 202 ),( 178 , 203 ),( 186 , 202 ),( 138 , 203 ),
( 149 , 203 ),( 156 , 202 ),( 163 , 203 ),( 170 , 202 ),( 177 , 202 ),
( 186 , 203 ),( 141 , 204 ),( 148 , 207 ),( 155 , 210 ),( 163 , 211 ),
( 171 , 210 ),( 179 , 207 ),( 185 , 203 ),( 103 , 141 ),( 101 , 160 ),
( 104 , 181 ),( 219 , 140 ),( 222 , 159 ),( 218 , 179 ),( 99 , 150 ),
( 92 , 144 ),( 88 , 149 ),( 90 , 160 ),( 94 , 174 ),( 99 , 187 ),
( 104 , 184 ),( 224 , 149 ),( 231 , 144 ),( 234 , 151 ),( 232 , 160 ),
( 230 , 173 ),( 224 , 185 ),( 219 , 184 ),( 104 , 181 ),( 108 , 199 ),
( 115 , 214 ),( 129 , 228 ),( 147 , 240 ),( 162 , 243 ),( 180 , 239 ),
( 196 , 228 ),( 207 , 215 ),( 215 , 199 ),( 219 , 178 ),( 101 , 144 ),
( 107 , 129 ),( 114 , 114 ),( 120 , 104 ),( 131 , 95 ),( 146 , 92 ),
( 160 , 93 ),( 174 , 95 ),( 188 , 96 ),( 201 , 103 ),( 210 , 114 ),
( 217 , 126 ),( 222 , 143 ),( 93 , 204 ),( 78 , 173 ),( 76 , 142 ),
( 82 , 101 ),( 99 , 70 ),( 129 , 46 ),( 158 , 44 ),( 188 , 45 ),
( 217 , 64 ),( 236 , 94 ),( 245 , 134 ),( 250 , 168 ),( 233 , 200 ),
( 145 , 175 ),( 139 , 182 ),( 135 , 190 ),( 178 , 176 ),( 185 , 183 ),
( 190 , 191 ),( 105 , 178 ),( 109 , 184 ),( 112 , 190 ),( 218 , 178 ),
( 214 , 183 ),( 211 , 189 ),( 159 , 186 ),( 159 , 193 ),( 165 , 186 ),
( 165 , 193 ),( 162 , 232 ),( 162 , 238 ),( 153 , 218 ),( 162 , 216 ),
( 173 , 219 )
```

Bild 5. Das „Normgesicht" dient als Grundlage zur Berechnung der Abweichungen in Bild 7

```
( 135 , 145 ),( 190 , 147 ),( 134 , 141 ),( 130 , 145 ),( 134 , 149 ),
( 140 , 145 ),( 134 , 141 ),( 190 , 143 ),( 185 , 146 ),( 190 , 150 ),
( 195 , 146 ),( 190 , 143 ),( 125 , 144 ),( 133 , 141 ),( 146 , 147 ),
( 177 , 142 ),( 187 , 139 ),( 201 , 145 ),( 125 , 144 ),( 133 , 150 ),
( 146 , 147 ),( 177 , 147 ),( 188 , 151 ),( 201 , 147 ),( 118 , 143 ),
( 134 , 139 ),( 146 , 144 ),( 176 , 147 ),( 188 , 141 ),( 201 , 147 ),
( 127 , 164 ),( 139 , 160 ),( 146 , 152 ),( 182 , 155 ),( 190 , 162 ),
( 201 , 164 ),( 156 , 145 ),( 156 , 153 ),( 155 , 163 ),( 154 , 171 ),
( 153 , 175 ),( 157 , 181 ),( 166 , 144 ),( 165 , 153 ),( 167 , 163 ),
( 168 , 171 ),( 171 , 177 ),( 167 , 181 ),( 152 , 176 ),( 147 , 178 ),
( 147 , 182 ),( 151 , 184 ),( 156 , 183 ),( 162 , 185 ),( 171 , 176 ),
( 176 , 179 ),( 176 , 184 ),( 172 , 184 ),( 168 , 183 ),( 162 , 185 ),
( 118 , 141 ),( 114 , 137 ),( 120 , 134 ),( 127 , 132 ),( 133 , 131 ),
( 148 , 134 ),( 176 , 135 ),( 174 , 134 ),( 184 , 131 ),( 192 , 132 ),
( 201 , 135 ),( 205 , 139 ),( 118 , 141 ),( 126 , 137 ),( 145 , 135 ),
( 148 , 134 ),( 176 , 135 ),( 190 , 134 ),( 200 , 139 ),( 205 , 139 ),
( 140 , 205 ),( 146 , 200 ),( 154 , 199 ),( 163 , 202 ),( 171 , 199 ),
( 179 , 202 ),( 180 , 206 ),( 140 , 205 ),( 145 , 205 ),( 157 , 204 ),
( 163 , 206 ),( 168 , 204 ),( 175 , 205 ),( 180 , 206 ),( 140 , 205 ),
( 145 , 205 ),( 157 , 204 ),( 163 , 206 ),( 168 , 204 ),( 175 , 205 ),
( 180 , 206 ),( 140 , 205 ),( 148 , 207 ),( 156 , 210 ),( 163 , 210 ),
( 171 , 210 ),( 179 , 208 ),( 180 , 206 ),( 98 , 145 ),( 101 , 154 ),
( 104 , 188 ),( 227 , 145 ),( 218 , 172 ),( 215 , 189 ),( 98 , 145 ),
( 92 , 144 ),( 90 , 156 ),( 91 , 164 ),( 94 , 173 ),( 98 , 184 ),
( 104 , 182 ),( 227 , 145 ),( 234 , 144 ),( 235 , 147 ),( 232 , 165 ),
( 227 , 177 ),( 222 , 181 ),( 216 , 182 ),( 104 , 188 ),( 113 , 206 ),
( 124 , 216 ),( 134 , 227 ),( 141 , 232 ),( 153 , 236 ),( 178 , 235 ),
( 192 , 225 ),( 202 , 218 ),( 211 , 202 ),( 215 , 189 ),( 98 , 145 ),
( 100 , 129 ),( 118 , 105 ),( 124 , 116 ),( 132 , 101 ),( 133 , 110 ),
( 169 , 99 ),( 167 , 109 ),( 175 , 102 ),( 179 , 108 ),( 188 , 101 ),
( 224 , 117 ),( 228 , 146 ),( 92 , 145 ),( 90 , 109 ),( 98 , 67 ),
( 116 , 48 ),( 128 , 49 ),( 149 , 41 ),( 178 , 46 ),( 196 , 40 ),
( 207 , 42 ),( 209 , 50 ),( 215 , 53 ),( 235 , 90 ),( 234 , 145 ),
( 146 , 173 ),( 145 , 175 ),( 141 , 176 ),( 174 , 172 ),( 180 , 176 ),
( 182 , 177 ),( 107 , 183 ),( 114 , 191 ),( 124 , 205 ),( 215 , 181 ),
( 206 , 189 ),( 197 , 203 ),( 161 , 189 ),( 158 , 197 ),( 165 , 189 ),
( 167 , 198 ),( 163 , 234 ),( 163 , 233 ),( 150 , 220 ),( 161 , 216 ),
( 171 , 221 )
```

Bild 6. Das Vergleichsgesicht für einen ersten Versuch liefert schon ganz brauchbare Ergebnisse

nem gesonderten Bogen die Reihenfolge mitschreibt.

Die scheinbar willkürliche Folge der Punkte ist damit zu erklären, daß wir ursprünglich vorhatten, eine Spline-Routine einzubauen, um die Bildqualität zu verbessern. Leider konnten wir Sie – wiederum aus Platzgründen – nicht implementieren. Aber da das Programm schon entsprechend vorbereitet ist, sollte es relativ leicht fallen, auf Wunsch eine entsprechende Routine einzubinden.

Tips zum Programm

Vor der ersten Benutzung sollte in Preferences die „Maus-Übersetzung" auf 4 gestellt werden, da jede Eingabe in der Prozedur DIGITIZER auf dem Bildschirm quittiert wird – bei kleineren Übersetzungen wird so leicht die Bildschirmgröße überschritten. Daher ist dieser Schritt natürlich bei sehr kleinen Vorlagen nicht nötig. Ansonsten kann die Vorlage (Fotos, Freihandzeichnungen usw.) fast jede Größe annehmen, da nach Ende der Eingabe eine automatische Skalierung durchgeführt wird (auch als gesonderter Menüpunkt vorhanden). Zweitens ist bei der Eingabe zu empfehlen, die Maus öfters zu justieren, da sonst bald Ungenauigkeiten auftreten. Aus dem selben Grunde empfiehlte sich auch, ein Fadenkreuz aus Pappe an der Maus anzubringen. Optimal ist hier natürlich ein Digitalisierer, der die Vorlagen direkt abtastet.

Mit dem Hilfsprogramm in *Bild 4* werden Norm- und Vergleichsgesicht eingegeben. Die Koordinaten unserer Beispielgesichter finden Sie in *Bild 5 und 6* (wir bitten um Nachsicht, das Vergleichsgesicht stammt aus einer selbstgefertigten Zeichnung, die einen bekannten Sportler darstellen sollte). Nach der Eingabe können Sie das Hilfsprogramm wieder löschen.

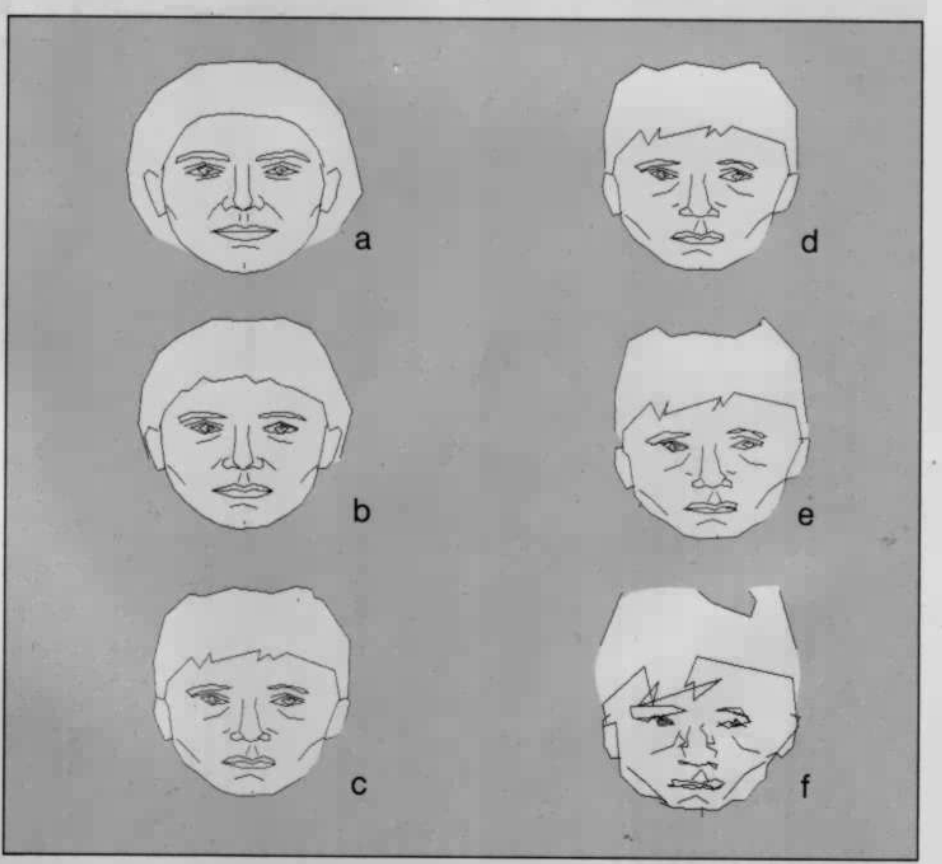


Bild 7. Mit verschiedenen Karikaturfaktoren läßt sich das Bild eines Gesichtes bis zur Unkenntlichkeit verändern: Links oben das „Normgesicht" (a), darunter eine 50-50-Mischung aus Norm- und Vergleichsgesicht (b), links unten das Vergleichsgesicht selbst (c); rechts Karikaturen mit den Faktoren 0,1 (d), 0,3 (e) und 1,2 (f)

Dipl.-Ing. Jens Kardoeus

Die Zeit ist reif

Timer in Pascal

Daß ein PC ausgerechnet für so etwas Elementares wie Zeitmessung zunächst nicht ohne weiteres geeignet ist, ärgert viele Anwender aus dem naturwissenschaftlich-technischen Bereich. Ein komfortables Instrumentarium in Turbo-Pascal ist da sehr wünschenswert.

Timer finden insbesondere in der Automatisierungs-Technik einen großen Anwendungsbereich. Dort werden sie beispielsweise zur Ein- und Ausschaltverzögerung oder zum Generieren von Impulsen eingesetzt. Was bei SPS-Sprachen (SPS steht für speicherprogrammierbare Steuerungen) schon lange zur Standard-Syntax gehört, ist in einer Hochsprache wie Pascal nur mit Einschränkungen und Kompromissen zu programmieren. Das in diesem Artikel beschriebene Programm-Modul TIMER.PAS zeigt eine ebenso einfache wie effektive Lösung in Turbo-Pascal unter MS-DOS.

Bei der Programmierung von Timer-Funktionen bieten sich zwei Möglichkeiten an: Entweder man benutzt den MS-DOS-Funktionsaufruf 2CH, um die Systemzeit abzufragen und laufend mit einer Anfangszeit zu vergleichen, oder man setzt einfach ein „DELAY (x)" an der geforderten Stelle im Programm ein. Die erste Methode erfordert bei der Berechnung der Zeitdiffe-

```
{ ************************************************************************ }
{ *  Modul  TIMER.PAS                                              2/10/87 * }
{ *                                                                        * }
{ *                    Timer, programmiert in PASCAL                       * }
{ *                     von Dipl. Ing. Jens Kardoeus                       * }
{ ************************************************************************ }

CONST
  Dseg_Save: INTEGER = 0; { zum Speichern des DS-Registers im Code-Segm.}
  Timer_Max = 100;        { max. verfügbare Timer }
TYPE
  regpack = RECORD CASE BYTE OF
            1: (ax,bx,cx,dx,bp,si,di,ds,es,flags: INTEGER);
            2: (al,ah,bl,bh,cl,ch,dl,dh: BYTE);
            END;
VAR
  Regs: regpack;
  Seg_1C: INTEGER ABSOLUTE $0000:$0072;
  Ofs_1C: INTEGER ABSOLUTE $0000:$0070;
  Old_Seg_1C, Old_Ofs_1C: INTEGER;
  Timer_Used: INTEGER;    { Anzahl der zur Zeit bearbeiteten Timer }
  Counter: ARRAY [1 .. Timer_Max] OF INTEGER;
  Control: ARRAY [1 .. Timer_Max] OF RECORD
                                     Impulse, Flank: BOOLEAN;
                                     Typ: CHAR
                                     END;

PROCEDURE timer_interrupt;  { Timer - Interrupt-Routine }
  BEGIN
  INLINE(
    $FB/                    { sti }
    $53/                    { push bx }
    $1E/                    { push ds }
    $2E/$FF/$36/Dseg_Save/  { push (cs:Dseg_Save) }
    $1F/                    { pop ds }
    $8B/$1E/Timer_Used/     { mov bx,Timer_Used }
    $4B/                    { dec bx }
    $D1/$E3/                { shl bx,1 }
  {1001}
    $83/$FB/$00/            { cmp bx,00h }
    $7C/$0F/                { jl  1003: }
    $83/$BF/Counter/$00/    { cmp [Counter+bx],00h }
    $7E/$04/                { jle 1002: }
    $FF/$8F/Counter/        { dec [Counter+bx] }
  {1002:}
    $4B/                    { dec bx }
    $4B/                    { dec bx }
    $EB/$EC/                { jmp 1001: }
  {1003:}
    $1F/                    { pop ds }
    $5B/                    { pop bx }
    $5C/                    { pop sp          pushed durch Turbo-Pascal }
    $5D/                    { pop bp          pushed durch Turbo-Pascal }
    $CF)                    { iret }
  END;
```

Bild 1. Timer-Interrupt-Routine als INLINE-Statement in einem Turbo-Pascal-Programm

```
PROCEDURE timer_init;          { Timer initialisieren }
  VAR i: INTEGER;
  BEGIN
  Timer_Used := 0;
  FOR i := 1 TO Timer_Max DO
    BEGIN
    Counter[i] := -1;
    WITH Control[i] DO
      BEGIN
      Impulse := FALSE;
      Flank := FALSE;
      Typ := ' '
      END
    END
  END;

PROCEDURE timer_set;           { Timer - Interrupt setzen }
  BEGIN
  timer_init;                  { Initialisieren der Timer }
  Old_Ofs_1C := Ofs_1C;        { Ofset vom Interrupt-Vektor sichern }
  Old_Seg_1C := Seg_1C;        { Segment vom Interrupt-Vektor sichern }
  Dseg_Save := DSEG;           { Datensegment sichern }
  PORT[$43] := $36;            { Umprogrammierung des Kanals 0 }
  PORT[$40] := $0B;            { des Timer-Bausteins 8253 auf }
  PORT[$40] := $E9;            { ein Zeitnormal von 50 msec }
  WITH Regs DO
    BEGIN
    ax := $251C;
    ds := CSEG;
    dx := OFS(timer_interrupt);
    MSDOS(Regs)                { Umlegen des Interrupt-Vektors 1C }
    END;                       { auf Procedure 'timer_interrupt' }
  END;

PROCEDURE timer_reset;         { Timer - Interrupt rücksetzen }
  BEGIN
  PORT[$43] := $36;            { Rückprogrammierung des Kanals 0 }
  PORT[$40] := $FF;            { des Timer-Bausteins 8253 auf das }
  PORT[$40] := $FF;            { alte Zeitnormal von 54,9 msec }
  WITH Regs DO
    BEGIN
    ax := $251C;
    ds := Old_Seg_1C;
    dx := Old_Ofs_1C;
    MSDOS(Regs);               { Umlegen des Interrupt-Vektors 1C }
    END;                       { auf BIOS - Dummy }
  END;
```

Bild 2. Initialisierung, Setzen und Rücksetzen des Timer-Interrupts

```
{$I TIMER.PAS}
VAR i: INTEGER;
  BEGIN
  CLRSCR;
  GOTOXY (20,10);
  WRITE ('Laufzeit in 0.1 sec. eingeben : ');
  READ (i);
  timer_set;                   (* Timer - Interrupt setzen *)
  Timer_Used := 1;             (* Anzahl der zu bearbeitenden Timer *)
  timer_start (1,i);           (* Timer 1 mit i/10 sec. starten *)
  REPEAT
    GOTOXY (52,10);
    WRITE (timer_value(1):3);  (* Timer - Wert ausgeben *)
  UNTIL timer_value (1) = 0;   (* Timer abgelaufen ? *)
  timer_reset                  (* Timer - Interrupt ruecksetzen *)
  END.
```

Bild 4. Ein einfaches Anwendungsbeispiel für einen Timer

```
PROCEDURE timer_start (number,time: INTEGER);
  { Starten eines Timers }
  BEGIN
  Counter[number] := time SHL 1
  END;

FUNCTION timer_value (number: INTEGER): INTEGER;
  { Wert eines Timers abfragen }
  BEGIN
  timer_value := Counter[number] DIV 2
  END;
```

Bild 3. Starten und Abfragen des Timers

```
FUNCTION timer (number: INTEGER): BOOLEAN;
  { Abfragen eines Timers }
  BEGIN
  timer := FALSE;
  IF number <= Timer_Used THEN
    CASE control[number].Typ OF
      'i': IF Counter[number] > 0 THEN timer := TRUE;
      'e': IF Counter[number] = 0 THEN timer := TRUE;
      'a': IF Counter[number] <>0 THEN timer := TRUE
    END
  END;
```

Bild 5. Die Funktion timer gibt den Zustand eines Timers an

renz relativ komplexe Algorithmen, und schon beim parallelen Lauf mehrerer Timer erstickt das Programm quasi in der Zeiterfassung. Die zweite Methode ist vom Aufwand her sicherlich die einfachere, blokkiert das Programm aber während des DELAYs völlig.

Die folgende Lösung basiert auf der Nutzung des Timer-Interrupts 1CH. Der Vektor dieses BIOS-Interrupts zeigt nach dem Booten auf ein Dummy (iret) im BIOS und ist für Anwender-Programmierung reserviert. Er wird vom Timer-Baustein 8253 über den Timer-Kanal 0 insgesamt 18,2mal pro Sekunde angesprungen.

Hier soll der Vektor auf die Prozedur timer_interrupt (*Bild 1*) gerichtet werden, so daß diese mit jedem Timer-Interrupt einmal ausgeführt wird. Die Befehle sind mittels einer INLINE-Anweisung im Maschinencode geschrieben, um eine schnelle Abarbeitung zu erreichen. In dieser Prozedur werden pro Aufruf alle Elemente des Arrays Counter [1...Timer_Used] (siehe Deklarationsteil) einmal dekrementiert, bis sie jeweils den Wert 0 erreicht haben. Die Variable Timer_Used kann während des Programmablaufes verändert und damit an die Anzahl der zur Zeit benutzten Timer angepaßt werden, wodurch die Bearbeitungszeit der Interrupt-Routine möglichst gering gehalten werden kann. Der Wert darf aber auf keinen Fall größer als der von Timer_Max sein, sonst könnten Programmteile von der Interrupt-Routine zerstört werden. Das Umlegen des Interrupt-Vektors erfolgt in der Prozedur timer_set mit dem MS-DOS-Funktionsaufruf 25H (*Bild 2*). Vor der Beendigung des Anwender-Programms muß der Vektor mit der Prozedur timer_reset unbedingt wieder auf das BIOS gerichtet werden, sonst erfolgt unweigerlich ein Systemabsturz.

Wie eingangs bereits erwähnt, erfolgt der Timer-Interrupt 18,2mal pro Sekunde. Daraus folgt ein Zeitnormal von ca. 55 ms. Eine sinnvolle Zeitmessung ist aber nur in Intervallen von ganzen Zehnerpotenzen möglich. Darum wird in der Prozedur timer_set der Kanal 0 des Timer-Bausteines auf ein Zeitnormal von exakt 50 ms umprogram-

```
PROCEDURE timer_si (number,time: INTEGER; start,reset: BOOLEAN);
  { Starten eines Timers als Impuls }
  BEGIN
  WITH Control[number] DO
    BEGIN
    Impulse := start AND NOT Flank;
    IF Impulse THEN
      BEGIN Counter[number] := time SHL 1; Typ := 'i'; Flank := TRUE END;
    IF NOT start THEN
      BEGIN Flank := FALSE; Counter[number] := -1 END;
    IF reset THEN
      Counter[number] := -1
    END
  END;
```

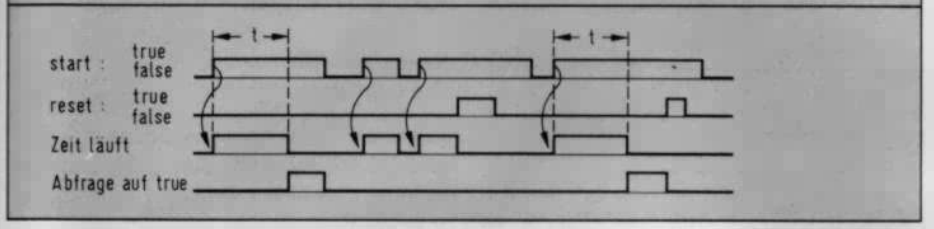


Bild 6. Starten eines Timers als Impuls

```
PROCEDURE timer_sv (number,time: INTEGER; start,reset: BOOLEAN);
  { Starten eines Timers als verlängerten Impuls }
  BEGIN
  WITH Control[number] DO
    BEGIN
    Impulse := start AND NOT Flank;
    IF Impulse THEN
      BEGIN Counter[number] := time SHL 1; Typ := 'i'; Flank := TRUE END;
    IF NOT start THEN
      Flank := FALSE;
    IF reset THEN
      Counter[number] := -1
    END
  END;
```

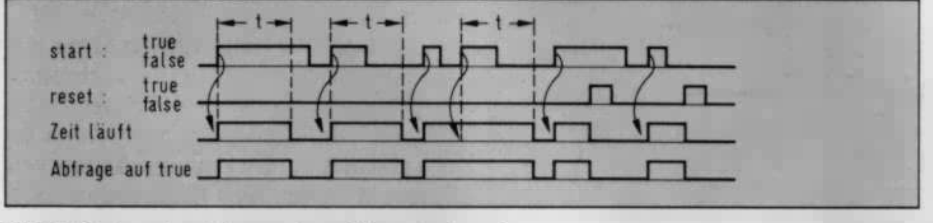


Bild 7. Starten eines Timers als verlängerter Impuls

```
PROCEDURE timer_se (number,time: INTEGER; start,reset: BOOLEAN);
  { Starten eines Timers als Einschaltverzögerung }
  BEGIN
  WITH Control[number] DO
    BEGIN
    Impulse := start AND NOT Flank;
    IF Impulse THEN
      BEGIN Counter[number] := time SHL 1; Typ := 'e'; Flank := TRUE END;
    IF NOT start THEN
      BEGIN Flank := FALSE; Counter[number] := -1 END;
    IF reset THEN
      Counter[number] := -1
    END
  END;
```

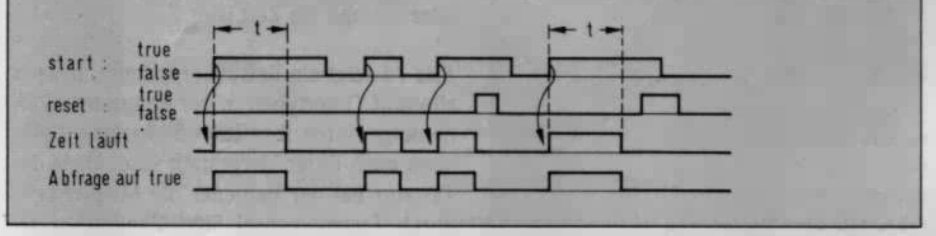


Bild 8. Starten eines Timers als Einschaltverzögerung

miert. Auch dies wird in der Prozedur timer_reset rückgängig gemacht. Bei der Umprogrammierung des Bausteines ist zu bedenken, daß über den Kanal 0 zwei weitere Routinen ausgelöst werden. Die eine steuert den Nachlauf der Diskettenlaufwerke und die andere bearbeitet die Systemuhr. Während es ziemlich unerheblich ist, ob ein Diskettenlaufwerk nach einem Zugriff 2,0 oder 1,8 s nachläuft, so kann eine um 10% zu schnell laufende Uhr doch schon recht ärgerlich sein. An dieser Stelle sei aber bemerkt, daß die Systemuhr ohnehin nicht die genaueste ist und in jedem zweiten PC mittlerweile eine externe Uhr steckt, die sich direkt ansprechen läßt.

Die Schnittstelle zum Anwender-Programm bilden die Prozedur timer_start (number,time: INTEGER) und die Funktion timer_value (number: INTEGER): INTEGER (*Bild 3*). Der Parameter number gibt die Nummer des zu startenden bzw. abzufragenden Timers an. Ein Timer wird von der Interrupt-Routine nur bearbeitet, wenn seine Nummer im Bereich 1 bis Timer_Used liegt. Der Parameter time übergibt die Zeit, mit der der Timer gestartet werden soll. Seine Einheit beträgt 0,1 s, d. h. wird ein Timer mit dem Wert 20 für time gestartet, so beträgt seine Laufzeit 2 s. *Bild 4* zeigt ein einfaches Beispiel für die Verwendung eines Timers.

Einsatzmöglichkeiten von Timern

Fünf der häufigsten Einsatztypen mit ihren Impulsdiagrammen und der Realisierung in Pascal sollen vorgestellt werden. Jede Prozedur enthält die Parameter number und time sowie die boolschen Variablen start und reset. Insbesondere bei zyklischer Programmbearbeitung ist es sinnvoll, mit einer Flankensteuerung zu arbeiten, daher werden alle Timer mit einer Signalflanke (von FALSE nach TRUE bzw. von TRUE nach FALSE) der Variablen start gestartet. Das Rücksetzen erfolgt zustandsgesteuert mit dem Parameter reset, d. h. solange dieser den Wert TRUE hat, kann der entsprechende Timer nicht gestartet werden. Jeder Timer wird mit der boolschen Funktion timer (number: INTEGER): BOOLEAN (*Bild 5*) auf die Zustände TRUE oder FALSE hin abgefragt.

Starten eines Timers als Impuls

Mit einer Flanke von FALSE nach TRUE wird die Zeit gestartet (*Bild 6*). Sie läuft mit der programmierten Zeitdauer time ab, solange die Variable start den Wert TRUE hat. Wechselt der Wert nach FALSE, bevor die Zeit abgelaufen ist, wird abgebrochen. Die Abfrage des Timers liefert den Wert TRUE, solange die Zeit läuft.

```
PROCEDURE timer_ss (number,time: INTEGER; start,reset: BOOLEAN);
  { Starten eines Timers als speichernde Einschaltverzögerung }
  BEGIN
  WITH Control[number] DO
    BEGIN
    Impulse := start AND NOT Flank;
    IF Impulse THEN
      BEGIN
      IF Counter[number] <> 0 THEN
        BEGIN Counter[number] := time SHL 1; Typ := 'e' END;
      Flank := TRUE
      END;
    IF NOT start THEN
      Flank := FALSE;
    IF reset THEN
      Counter[number] := -1
    END
  END;
```

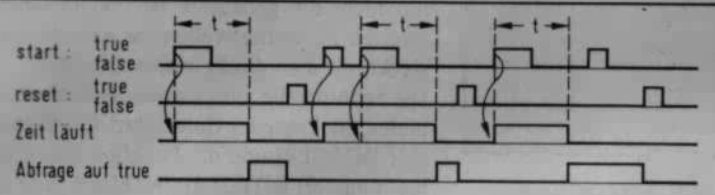


Bild 9. Starten eines Timers als speichernde Einschaltverzögerung

```
PROCEDURE timer_sa (number,time: INTEGER; start,reset: BOOLEAN);
  { Starten eines Timers als Ausschaltverzögerung }
  BEGIN
  WITH Control[number] DO
    BEGIN
    Impulse := NOT start AND Flank;
    IF Impulse THEN
      BEGIN Counter[number] := time SHL 1; Flank := FALSE END;
    IF start THEN
      BEGIN Counter[number] := -1; Typ := 'a'; Flank := TRUE END;
    IF reset THEN
      Counter[number] := 0
    END
  END;
```

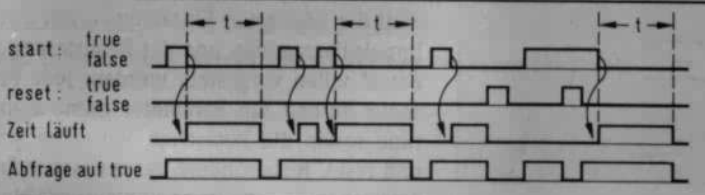


Bild 10. Starten eines Timers als Ausschaltverzögerung

```
{$I TIMER.PAS}
VAR c: CHAR;
    i: INTEGER;
  BEGIN
  CLRSCR;
  timer_set;
  Timer_Used := 5;
  c := ' ';
  GOTOXY (1,25);
  WRITE ('Starte Timer mit Tastenwechsel nach "s" -- Rücksetze mit "r"');
  GOTOXY (2,5);
  REPEAT
    DELAY (200);
    IF KEYPRESSED THEN READ (KBD,c);
    timer_si (1,20,c='s',c='r');
    timer_sv (2,20,c='s',c='r');
    timer_se (3,20,c='s',c='r');
    timer_ss (4,20,c='s',c='r');
    timer_sa (5,20,c='s',c='r');
    FOR i := 1 TO 5 DO
      BEGIN
      GOTOXY (WHEREX-1,3+2*i);
      IF timer(i) THEN WRITE (CHR(219)) ELSE WRITE (CHR(95));
      END;
    GOTOXY (WHEREX+1,WHEREY)
  UNTIL WHEREX = 80;
  timer_reset
  END.
```

Bild 11. Alle fünf Timertypen im Einsatz

Starten eines Timers als verlängerter Impuls

Mit einer Flanke von FALSE nach TRUE wird die Zeit gestartet (*Bild 7*). Sie läuft mit der programmierten Zeitdauer time ab, unabhängig vom Zustand der Variablen start. Wechselt der Wert von FALSE nach TRUE, während die Zeit läuft, wird der Timer mit dem programmierten Zeitwert nachgetriggert. Die Abfrage liefert den Wert TRUE, solange die Zeit läuft.

Starten eines Timers als Einschaltverzögerung

Mit einer Flanke von FALSE nach TRUE wird die Zeit gestartet (*Bild 8*). Sie läuft mit der programmierten Zeitdauer time ab, solange die Variable start den Wert TRUE hat. Wechselt der Wert nach FALSE, bevor die Zeit abgelaufen ist, wird abgebrochen. Die Abfrage des Timers liefert den Wert TRUE, nachdem die Zeit vollständig abgelaufen ist und solange die Variable noch den Wert TRUE besitzt.

Starten eines Timers als speichernde Einschaltverzögerung

Mit einer Flanke von FALSE nach TRUE wird die Zeit gestartet (*Bild 9*). Sie läuft mit der programmierten Zeitdauer time ab, unabhängig vom Zustand der Variablen start. Wechselt der Wert von FALSE nach TRUE, während die Zeit läuft, wird der Timer mit dem programmierten Zeitwert nachgetriggert. Nach Ablauf der Zeit muß der Timer mit reset zurückgesetzt werden, bevor er neu gestartet werden kann. Die Abfrage des Timers liefert den Wert TRUE, nachdem die Zeit vollständig abgelaufen ist, unabhängig vom Zustand der Variablen start, bis der Timer mit „reset" zurückgesetzt wird.

Starten eines Timers als Ausschaltverzögerung

Mit einer Flanke von TRUE nach FALSE wird die Zeit gestartet (*Bild 10*). Sie läuft mit der programmierten Zeitdauer time ab, solange die Variable start den Wert FALSE hat. Wechselt der Wert nach TRUE, bevor die Zeit abgelaufen ist, wird abgebrochen. Die Abfrage des Timers liefert den Wert TRUE, wenn start den Wert TRUE besitzt oder solange die Zeit läuft.

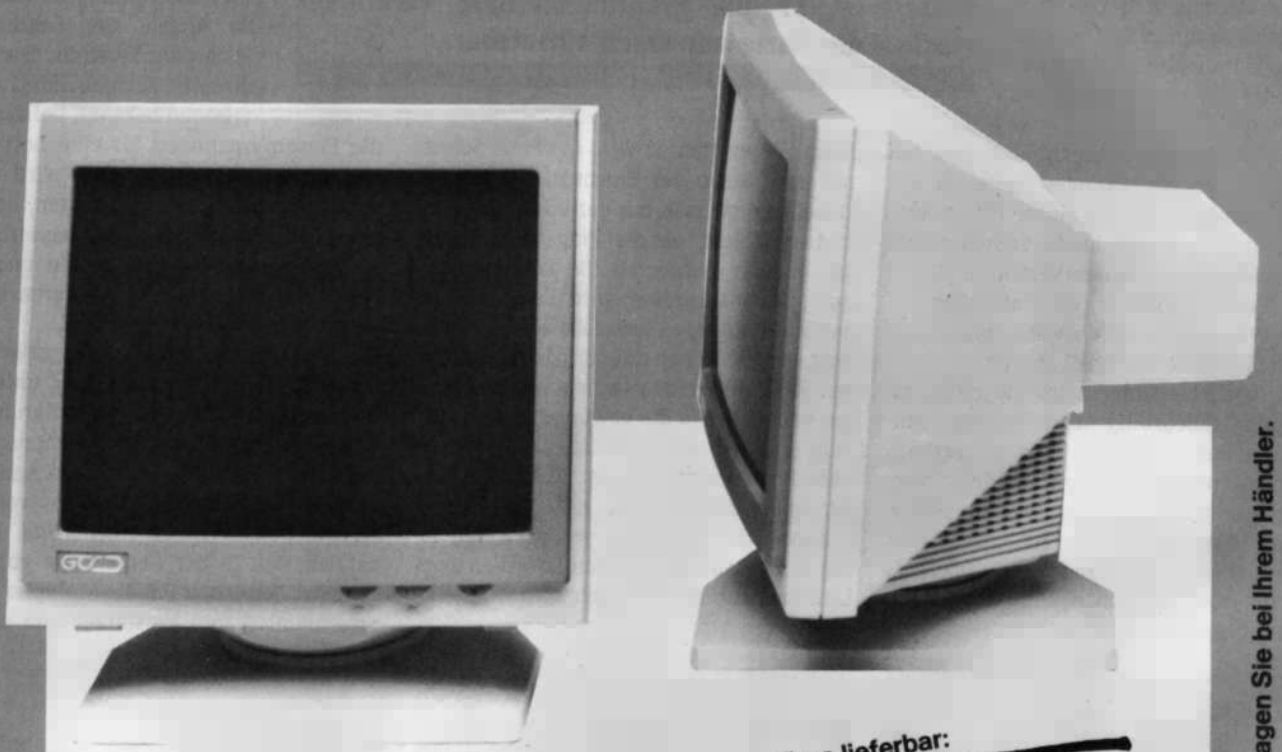
Bild 11 zeigt ein Beispielprogramm, in dem alle fünf Timertypen in der erläuterten Reihenfolge durch ihre Impulsdiagramme von oben nach unten dargestellt sind. Über die Tastatur hat der Bediener die Möglichkeit, durch Tastenwechsel Signalflanken zu erzeugen. □

HighTech

GCS T-1418 / T-1428

HighTech-Monitore von GEOSOFT

Mit weniger sollten Sie sich nicht zufrieden geben.



CRT:
14"

Display:
235 x 165 mm

Schwenkbereich:
von rechts nach links –180°
in der Höhe – 30°

Bandweite:
20 MHZ (– 3 dB)

Auflösung:
1000 Linien (innen)
900 Linien (außen)

Farben:
Amber / Paperwhite

Bedienungs-Elemente:
vorn – Ein-/Ausschalter
– Kontrast
– Helligkeit

hinten – Schalter für normale oder inverse Darstellung

Frequenz:
T-1418 – 18.432 KHZ
T-1428 – 15.75 / 18.432 KHZ

Takt-Netzteil:
115 V, 60 HZ /
220 V, 50 HZ umschaltbar

Abmessungen:
345 (H) x 310 (T) x 324 (B) mm

In Kürze lieferbar:

GCS MULTISYNC FARBE
GCS MULTISYNC MONOCHROM
GCS VGA FARBE
GCS MONOCHROM 19"

GEOSOFT

GEOSOFT
Brender & Kraft GmbH
Am Ungersgarten 20 — Telefon 0 62 02 / 2 59 80
D-6831 Plankstadt — Telex 4 67 011 geos d

Distributoren in ganz Europa gesucht.

Vertrieb nur über den guten Fachhandel. Fragen Sie bei Ihrem Händler.

Ulrich Gärtner

Professioneller A/D-Umsetzer für den PC

12-Bit-Auflösung mit modernen Bausteinen realisiert

Eine Steckkarte für den PC zur Messung von analogen Werten läßt sich vielseitig einsetzen. Die AD12-Karte hat eine Auflösung von 12 Bit bei 25 µs Umsetzzeit. Bei 8 Bit Auflösung beträgt die Umsetzzeit 17 µs. Programmierbare Verstärker machen die Karte universell einsetzbar.

Um analoge Signale in einem PC zu verarbeiten, müssen sie in für den Rechner erfaßbare Zahlen (Digitalwerte) umgesetzt werden. Diese Werte können im Rechner weiterverarbeitet werden, um z. B. eine Fouriertransformation durchzuführen oder um eine Regelung zu realisieren.

Als A/D-Umsetzer werden Einheiten bezeichnet, die ein analoges Signal in ein proportionales digitales Signal (Wort) umwandeln. Man unterscheidet zwischen drei prinzipiell verschiedenen Verfahren [1]:

– das Parallelverfahren (word at a time)
– das Wägeverfahren (digit at a time)
– das Zählverfahren (level at a time)

Beim Parallelverfahren wird die Eingangsspannung gleichzeitig mit n Referenzspannungsquellen verglichen und festgestellt, zwischen welchen beiden sie liegt. Man erhält damit den digitalen Wert in einem Schritt. Das Verfahren ist zwar sehr schnell, aber der Aufwand ist sehr hoch, da man für jeden möglichen Wert einen Komparator benötigt.

Beim Wägeverfahren wird in einem Schritt nur eine Stelle des Ergebnisses ermittelt. Dabei beginnt man mit der höchsten Stelle und stellt fest, ob die Eingangsspannung größer oder kleiner als die Referenzspannung ist. Ist sie größer, wird die erste Stelle auf Eins gesetzt, anderenfalls auf Null. Den Rest vergleicht man mit der nächstniedrigeren Stelle. Es werden so viele Vergleichstellen benötigt, wie die gewünschte Genauigkeit erfordert und natürlich auch ebenso viele Referenzspannungen. Bei diesem Verfahren ist sichergestellt, daß man das Ergebnis nach n Schritten vorliegen hat, somit kann man auch die Zeit zwischen zwei Umsetzungen direkt angeben, d. h. sie ist konstant.

Das Zählverfahren ist das einfachste Verfahren. Dabei zählt man ab, wie oft die Referenzspannung der niedrigsten Stelle addiert werden muß, um die Eingangsspannung zu erhalten. Die Anzahl der Schritte ist gleich dem Ergebnis. Dies Verfahren ist auch als Integrationsverfahren bekannt, bei dem erst die Eingangsspannung für eine bestimmte Zeit aufintegriert wird, dann wird umgeschaltet und mit einer konstanten, bekannten Referenzspannung abintegriert. Die Zeit zwischen dem Umschalten und dem Nulldurchgang beim Abintegrieren ist der Eingangsspannung proportional.

Die Karte AD12 (*Bild 1*) ist in zwei Teile, den Digital- und den Analogteil, unterteilt. Im Digitalteil wird die Decodierung der Karte sowie die Zwischenspeicherung von Daten vorgenommen (*Bild 2*). Im Analogteil befinden sich drei Baugruppen vor dem eigentlichen A/D-Umsetzer (*Bild 3*). Es handelt sich in der Reihenfolge vom Eingang her betrachtet um die Multiplexer, die programmierbaren Verstärker und die Abtast-Halte-Verstärker.

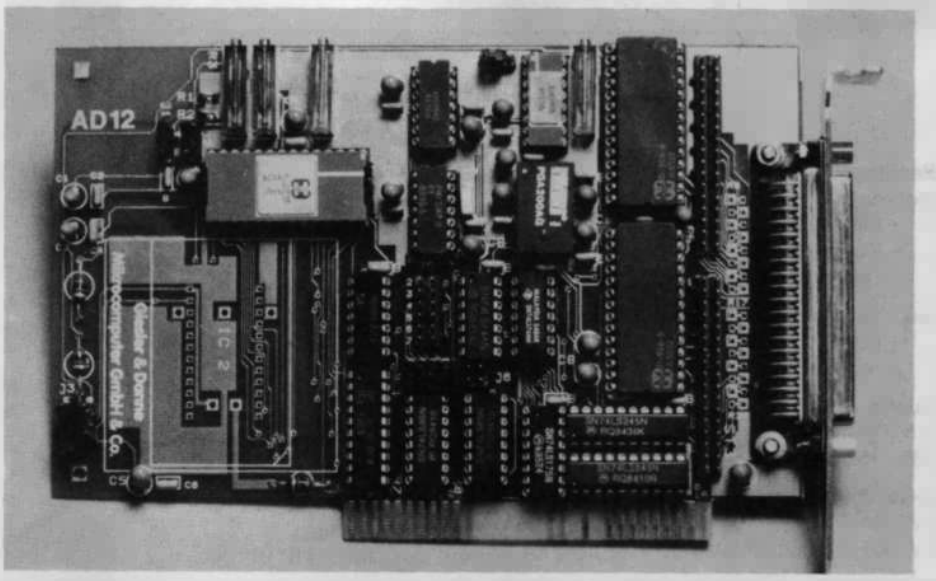


Bild 1. So sieht die Karte voll bestückt bestückt aus. Bei der Bestückung ist der Hinweis in der Stückliste zu beachten

Der Analogteil im Detail

Je nach Art des Eingangsignals wird die Karte entweder mit dem Multiplexer AD7506 oder AD7507 bestückt. Der AD7506 wird bei massebezogenen Eingängen und der AD7507 bei differentiellen Eingängen verwendet. Bestückt man die Karte mit dem AD7507, muß man auch die Widerstände R9...R24 einlöten. Die Multiplexer sind mit vier Eingängen ausgestattet, die von einem Mikroprozessor steuerbar sind. Über diese Eingänge wird der Schaltzustand der Multiplexer gesteuert. Die anliegenden Daten werden übernommen, wenn der En-Eingang auf H-Pegel liegt. Dieser Eingang wirkt wie ein Chip-Select. An die Analog-Eingänge darf eine Spannung

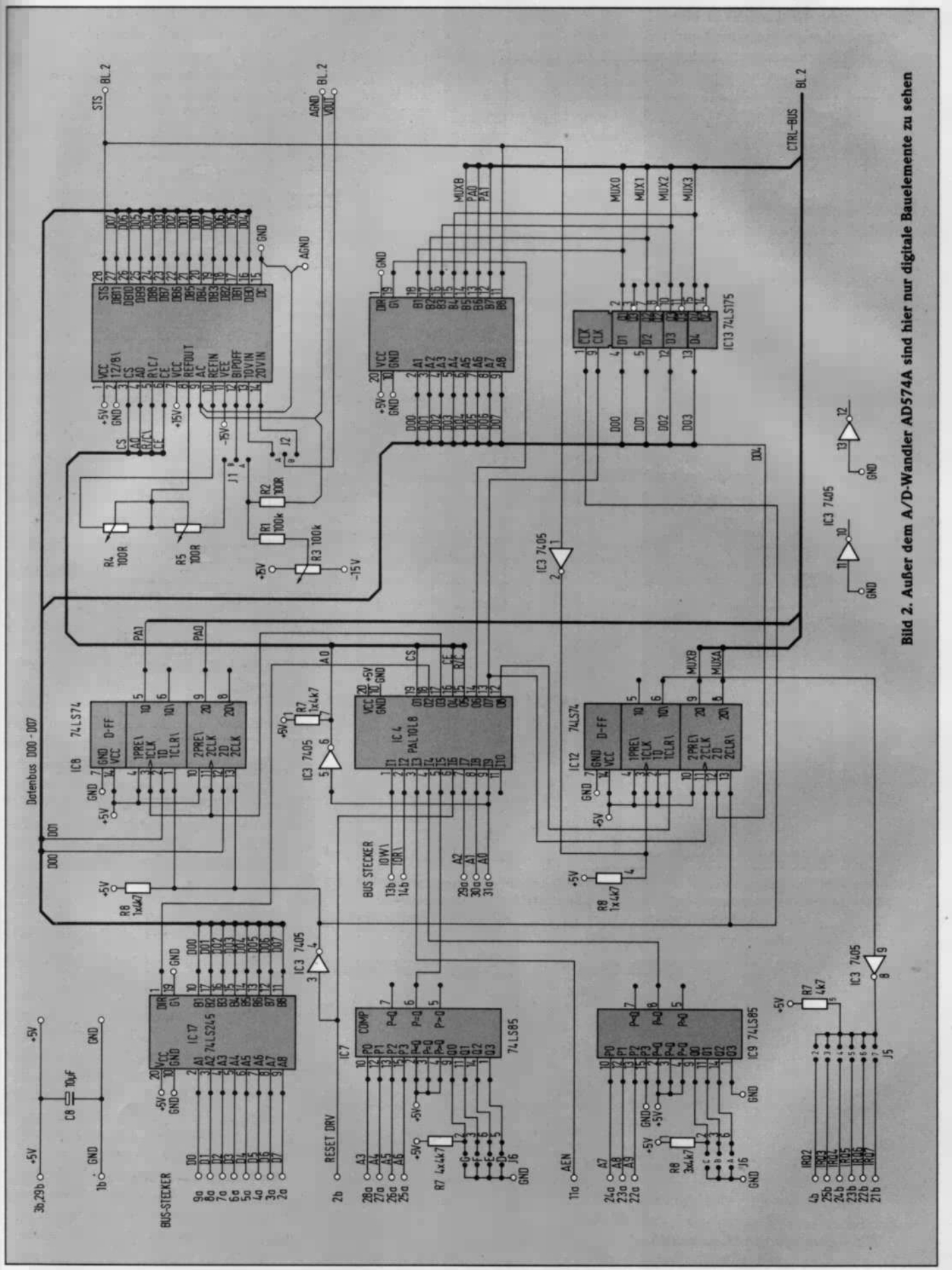


Bild 2. Außer dem A/D-Wandler AD574A sind hier nur digitale Bauelemente zu sehen

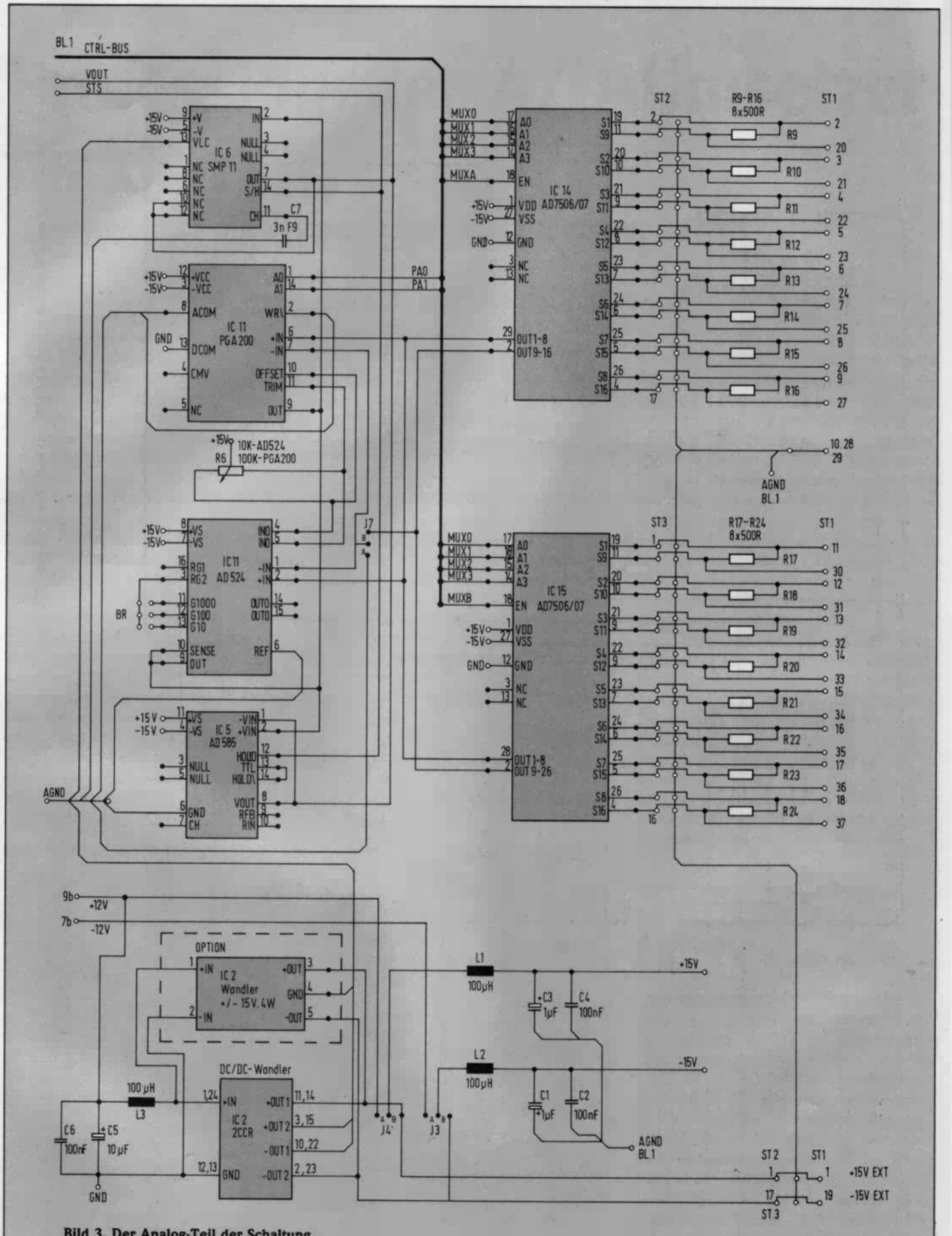


Bild 3. Der Analog-Teil der Schaltung

von maximal 25 Volt gelegt werden, was bei der AD12-Karte in allen Bereichen eingehalten wird.
Die Belegungen der Stecker und Pfostenleisten sind den *Bildern 4...9* zu entnehmen.
Die Verstärker auf der Karte dienen zur Normierung von Eingangssignalen, falls sie nicht extern verstärkt werden. Durch die Verstärker wird der Umsetzungsbereich des nachgeschalteten A/D-Wandlers voll ausgenutzt. Außerdem darf der Anwender Sensoren mit geringer Ausgangsspannung (z. B. Temperatursensoren oder Meßbrükken) direkt anschließen. Falls man die Karte mit dem Verstärker-Baustein PGA200 bestückt, kann man die Verstärkung der einzelnen Kanäle unterschiedlich einstellen.
Es ergibt sich bei einem Verstärkungsfaktor von 100 der in *Tabelle 1* dargestellte Eingangsspannungsbereich. Es ist deutlich zu sehen, daß die Stellung der Brücken J1/J2 auch von dem verwendeten Verstärkungsfaktor abhängt, bzw. sich auf einen Verstärkungsfaktor von 1 bezieht.

Verstärkung per Software einstellbar

Der PGA200 ist ein durch Software programmierbarer Verstärker mit vier Verstärkungstufen (1, 10, 100, 1000). Seine Eingänge sind TTL-kompatibel und können direkt vom Mikroprozessor gesteuert werden.
Über die Eingänge A0 und A1 wird die Verstärkung eingestellt (*Tabelle 2*). Der Eingang $\overline{WR}$ (Pin 2) steuert ein internes D-Flipflop (D-Latch). Ist $\overline{WR}$ = 0, dann ist das D-Flipflop transparent und die Verstärkung wird durch den Zustand der Signale A0 und A1 bestimmt. Wenn $\overline{WR}$ = 1 ist, dann speichert das D-Latch den vorherigen Zustand von A0 und A1.

Programmierbarer Verstärker AD524

Der AD524 ist ein durch Hardware programmierbarer Verstärker mit vier Verstärkungsstufen (1, 10, 100, 1000). Die Verstärkung wird auf der Lötseite der Platine fest eingestellt. Für andere Verstärkungsfaktoren zwischen 1 und 1000 muß ein Widerstand zwischen RG1 (Pin 16) und RG2 (Pin 3) geschaltet werden.
Der Verstärker ist mit einer Eigangsschutzschaltung versehen, bei geringer Verstärkung, z. B. 10 oder weniger, arbeitet der Eingangswiderstand als ein Begrenzungselement, das in Serie mit dem Eingang liegt.

Der AD524 hat intern hochpräzise Widerstände, die eine Verstärkung von 1 (offen), 10 (Pin 13), 100 (Pin 12), 1000 (Pin 11) ermöglichen, wenn RG2 (Pin 3) mit dem entsprechenden Eingang verbunden wird. Sollen andere als die intern voreingestellten Verstärkungsfaktoren realisiert werden, gibt es zwei Methoden.
Bei der ersten Methode verwendet man einen externen Widerstand zwischen RG1 (Pin 16) und RG2 (Pin 3). Der Wert des Widerstands Rg wird wie folgt berechnet:

$$Rg = 40k\Omega/(G - 1)$$

Für gute Resultate sollte für Rg ein Spindeltrimmer oder eine Reihenschaltung aus einem Metallfilm-Widerstand und einem Spindeltrimmer gewählt werden. Es muß auf jeden Fall ein Widerstand mit geringer Temperaturdrift sein.
Bei der zweiten Methode werden die internen Widerstände in einer Parallelschaltung mit verwendet. Diese Technik minimiert die Temperaturabhängigkeit des Verstärkungsfaktors und die Justierarbeit (*Bild 10, Bild 11, Tabelle 3*).
Den Verstärkungsfaktor bestimmt man so:
G = (40 kΩ x (Rintern + Rextern)/ (Rintern × Rextern)) + 1

Tabelle 1: Eingangsspannungsbereich bei hundertfacher Verstärkung

Eingangsspannung	Stellung der Brücke J1/J2
0 bis 100 mV	0 bis 10 V
− 50 mV bis + 50 mV	− 5 V bis + 5 V
−100 mV bis +100 mV	−10 V bis +10 V

Tabelle 2: Über die Eingänge A0 und A1 wird die Verstärkung des PGA200 eingestellt

A1	A0	$\overline{WR}$	Verstärkung (PGA200)
X	X	↑	Bisheriger Verstärkungsfaktor wird beibehalten
0	0	0	1
0	1	0	10
1	0	0	100
1	1	0	1000

Tabelle 3: Die Verstärkung des AD524 wird mit Widerständen eingestellt

Verstärkungsfaktor	R_{Intern}
10	4444,44 kΩ
100	404,04 kΩ
1000	40,04 kΩ

+15V ext	1 o	o	AGND
MUX-Kanal 1	o	o	AGND
MUX-Kanal 9	o	o	AGND
MUX-Kanal 2	o	o	AGND
MUX-Kanal 10	o	o	AGND
MUX-Kanal 3	o	o	AGND
MUX-Kanal 11	o	o	AGND
MUX-Kanal 4	o	o	AGND
MUX-Kanal 12	o	o	AGND
MUX-Kanal 5	o	o	AGND
MUX-Kanal 13	o	o	AGND
MUX-Kanal 6	o	o	AGND
MUX-Kanal 14	o	o	AGND
MUX-Kanal 7	o	o	AGND
MUX-Kanal 15	o	o	AGND
MUX-Kanal 8	o	o	AGND
MUX-Kanal 16	o	o	AGND

Bild 4. Signalbelegung vom Pfostenstecker ST2 bei Bestückung mit dem AD7506 (32 massebezogene Kanäle)

MUX-Kanal 17	1 o	o	AGND
MUX-Kanal 25	o	o	AGND
MUX-Kanal 18	o	o	AGND
MUX-Kanal 26	o	o	AGND
MUX-Kanal 19	o	o	AGND
MUX-Kanal 27	o	o	AGND
MUX-Kanal 20	o	o	AGND
MUX-Kanal 28	o	o	AGND
MUX-Kanal 21	o	o	AGND
MUX-Kanal 29	o	o	AGND
MUX-Kanal 22	o	o	AGND
MUX-Kanal 30	o	o	AGND
MUX-Kanal 23	o	o	AGND
MUX-Kanal 31	o	o	AGND
MUX-Kanal 24	o	o	AGND
MUX-Kanal 32	o	o	AGND
-15V ext	o	o	AGND

Bild 5. Signalbelegung vom Pfostenstecker ST3 bei Bestückung mit dem AD7506 (32 massebezogene Kanäle):

+15V ext	1 o	o	AGND
MUX-Kanal 1 +	o	o	AGND
MUX-Kanal 1 -	o	o	AGND
MUX-Kanal 2 +	o	o	AGND
MUX-Kanal 2 -	o	o	AGND
MUX-Kanal 3 +	o	o	AGND
MUX-Kanal 3 -	o	o	AGND
MUX-Kanal 4 +	o	o	AGND
MUX-Kanal 4 -	o	o	AGND
MUX-Kanal 5 +	o	o	AGND
MUX-Kanal 5 -	o	o	AGND
MUX-Kanal 6 +	o	o	AGND
MUX-Kanal 6 -	o	o	AGND
MUX-Kanal 7 +	o	o	AGND
MUX-Kanal 7 -	o	o	AGND
MUX-Kanal 8 +	o	o	AGND
MUX-Kanal 8 -	o	o	AGND

Bild 6. Signalbelegung vom Pfostenstecker ST2 bei Bestückung mit dem AD7507 (16 differentielle Kanäle)

MUX-Kanal 9 +	1 o	o	AGND
MUX-Kanal 9 -	o	o	AGND
MUX-Kanal 10 +	o	o	AGND
MUX-Kanal 10 -	o	o	AGND
MUX-Kanal 11 +	o	o	AGND
MUX-Kanal 11 -	o	o	AGND
MUX-Kanal 12 +	o	o	AGND
MUX-Kanal 12 -	o	o	AGND
MUX-Kanal 13 +	o	o	AGND
MUX-Kanal 13 -	o	o	AGND
MUX-Kanal 14 +	o	o	AGND
MUX-Kanal 14 -	o	o	AGND
MUX-Kanal 15 +	o	o	AGND
MUX-Kanal 15 -	o	o	AGND
MUX-Kanal 16 +	o	o	AGND
MUX-Kanal 16 -	o	o	AGND
-15V ext	o	o	AGND

Bild 7. Signalbelegung vom Pfostenstecker ST3 bei Bestückung mit dem AD7507 (16 differentielle Kanäle)

Auch hier muß der externe Widerstand eine geringe Temperaturdrift aufweisen.

Die Abtast-Halte-Funktion

Da die Umwandlung bei allen A/D-Umsetzern eine gewisse Zeit in Anspruch nimmt, muß durch eine externe Beschaltung dafür gesorgt werden, daß das Eingangssignal während der Wandlung konstant bleibt. Dies ist besonders wichtig, wenn es sich um einen schnell veränderlichen Vorgang handelt. Der Abtast-Halte-Verstärker ist im Prinzip ein Analogschalter, der nur zwei Zustände kennt. Die Ausgangsspannung eines Abtast-Halte-Gliedes soll im eingeschalteten Zustand der Eingangsspannung folgen. In dieser Betriebsart verhält es sich also wie ein Analogschalter.
Im ausgeschalteten Zustand soll jedoch die Ausgangsspannung nicht Null werden, sondern die Spannung des Ausschaltzeitpunkts gehalten (gespeichert) werden. Durch die Abtast-Halte-Schaltung ist gewährleistet, daß die zu wandelnde Spannung während der Umsetzung konstant gehalten wird, während sie sonst der Eingangsspannung folgte und damit zu einem falschen Ergebnis führte. *Bild 12* veranschaulicht die beiden Betriebsarten. Es ist zu erkennen, daß der Abtast-Halte-Verstärker nichts mit einem Abtast-Halte-Glied zu tun hat, denn sonst müßte das Ausgangssignal eine Treppenfunktion sein. Die aus dem englischen abgeleitete Bezeichnung „sample and hold" ist also etwas irreführend. Die prinzipielle Anordnung eines Abtast-Halte-Verstärkers ist in *Bild 13* zu sehen. Das zentrale Bauelement ist der Kondensator C, der als Speicher dient. Wenn der Schalter S geschlossen ist, wird der Kondensator auf die Eingangsspannung aufgeladen. Damit dabei nicht die Eingangsspannungsquelle belastet wird, wird ein Impedanzwandler davorgeschaltet, der im Bild 13 als Spannungsfolger beschaltet ist. Er muß hohe Ausgangsströme liefern, um den Kondensator schnell umzuladen. Wenn der Schalter öffnet, soll die Spannung am Kondensator möglichst lange unverändert bleiben (in unserem Fall während der Abtastzeit). Deshalb ist in der Prinzipschaltung dem Kondensator auch noch ein Spannungsfolger nachgeschaltet,

+15V ext.	1o		
		o20	MUX-Kanal 9
MUX-Kanal 1	o	o	MUX-Kanal 10
MUX-Kanal 2	o	o	MUX-Kanal 11
MUX-Kanal 3	o	o	MUX-Kanal 12
MUX-Kanal 4	o	o	MUX-Kanal 13
MUX-Kanal 5	o	o	MUX-Kanal 14
MUX-Kanal 6	o	o	MUX-Kanal 15
MUX-Kanal 7	o	o	MUX-Kanal 16
MUX-Kanal 8	o	o	AGND
AGND	o	o	AGND
MUX-Kanal 17	o	o	MUX-Kanal 25
MUX-Kanal 18	o	o	MUX-Kanal 26
MUX-Kanal 19	o	o	MUX-Kanal 27
MUX-Kanal 20	o	o	MUX-Kanal 28
MUX-Kanal 21	o	o	MUX-Kanal 29
MUX-Kanal 22	o	o	MUX-Kanal 30
MUX-Kanal 23	o	o	MUX-Kanal 31
MUX-Kanal 24	o	o37	MUX-Kanal 32
-15V ext.	19o		

Bild 8. Signalbelegung des 37poligen D-Sub-Steckers ST1 bei Bestückung mit dem AD 7506 (32 massebezogene Kanäle)

+15V ext.	1o		
		o20	MUX-Kanal 1 -
MUX-Kanal 1 +	o	o	MUX-Kanal 2 -
MUX-Kanal 2 +	o	o	MUX-Kanal 3 -
MUX-Kanal 3 +	o	o	MUX-Kanal 4 -
MUX-Kanal 4 +	o	o	MUX-Kanal 5 -
MUX-Kanal 5 +	o	o	MUX-Kanal 6 -
MUX-Kanal 6 +	o	o	MUX-Kanal 7 -
MUX-Kanal 7 +	o	o	MUX-Kanal 8 -
MUX-Kanal 8 +	o	o	AGND
AGND	o	o	AGND
MUX-Kanal 9 +	o	o	MUX-Kanal 9 -
MUX-Kanal 10 +	o	o	MUX-Kanal 10 -
MUX-Kanal 11 +	o	o	MUX-Kanal 11 -
MUX-Kanal 12 +	o	o	MUX-Kanal 12 -
MUX-Kanal 13 +	o	o	MUX-Kanal 13 -
MUX-Kanal 14 +	o	o	MUX-Kanal 14 -
MUX-Kanal 15 +	o	o	MUX-Kanal 15 -
MUX-Kanal 16 +	o	o37	MUX-Kanal 16 -
-15V ext.	19o		

Bild 9. Signalbelegung des 37poligen D-Sub-Steckers ST1 bei Bestückung mit dem AD 7507 (16 differentielle Kanäle)

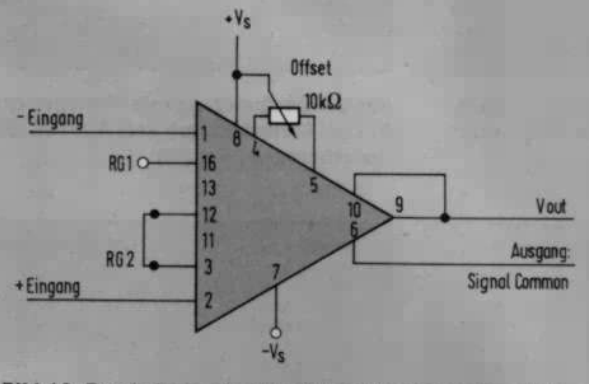


Bild 10. Beschaltung des AD 524 bei 100facher Verstärkung

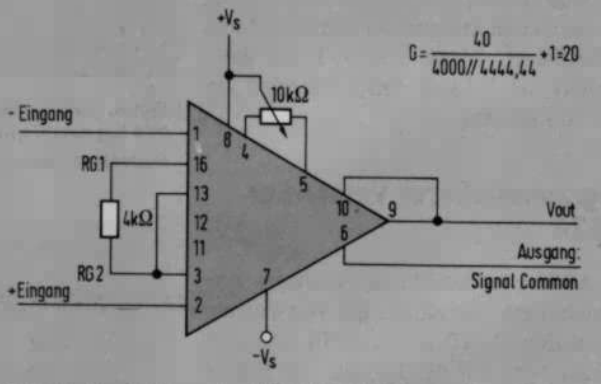
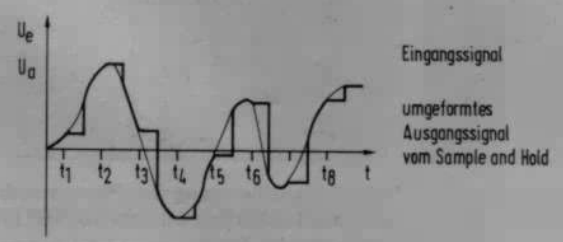


Bild 11. Beschaltung des AD 524 bei 20facher Verstärkung



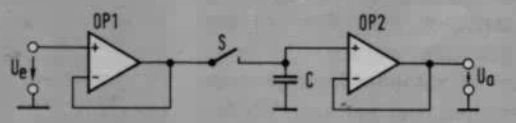


Bild 13. So ist ein Abtast-Halte-Verstärker aufgebaut

◁ **Bild 12. Signalverlauf bei einem Abtast-Halte-Verstärker**

Kennen Sie den Unterschied zwischen "preiswert" und "billig"?

Unsere Computer kosten soviel, wie es die Qualität erfordert - und so wenig, daß Sie sie sich bequem leisten können.

Unsere XTs, ATs, 386er und Portables sind **State of the Art.**

Natürlich berücksichtigen wir darüber hinaus die besondere Situation von Schülern, Lehrern, Studenten und Universitäten...

OSBORNE

...der Unterschied macht´s.

BRD

OSBORNE COMPUTER
GmbH
Dingolfinger Straße 6
8000 München 80
Tel. (089) 49 10 01
Telex 5 29 365
Fax (089) 49 70 40

Österreich

BASIC-Merton
Handelsgesellschaft mbH
Matthias-Schönerer-Gasse 11
1150 Wien
(2 22) 95 05 41
116 540 telebox "basic"

Schweiz

ARCO-DATA AG
Alte Steinhauser Str. 35
6330 Cham
(0 42) 41 78 41

Coupon

Zeigen Sie mir den Unterschied zwischen billig und **preiswert**! Informieren Sie mich über Ihre

mc 3/88

☐ XTs ☐ ATs ☐ 386er ☐ Portables

Name

Straße

PLZ/Ort

Telefon

OSBORNE COMPUTER GmbH, Dingolfinger Str. 6, 8000 München 80

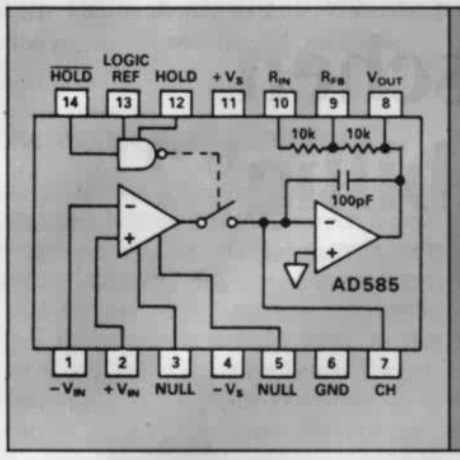


Bild 14. Anschlußbelegung und Schaltschema des AD585

```
PAL10L8                                                PAL DESIGN SPECIFICATION

PAL 10L8 fuer AD12-Karte

IOW IOR AEN SELA SELB RES A2  A1  A0 GND
NC  IFF MUX STAT RC   CE  PGA DIR CS VCC


/DIR  = SELA * SELB * /AEN * /IOR *  IOW * /A2                    ; Direction LS245
/STAT = SELA * SELB * /AEN * /IOR *  IOW * /A2 *  A1             ; Status
/CS   = SELA * SELB * /AEN *                     /A2 * /A1       ; Chip Select AD574
/RC   =                              IOR                         ; Read Control AD574
/CE   =                              IOR *  IOW                  ; Chip Enable AD574
/MUX  = SELA * SELB * /AEN *  IOR * /IOW * /A2 *  A1 * /A0       ; Multiplexer
/PGA  = SELA * SELB * /AEN *  IOR * /IOW * /A2 *  A1 *  A0       ; PGA
/IFF  = SELA * SELB * /AEN *  IOR * /IOW *  A2 * /A1 * /A0       ; Interrupt-Flipflop

      + RES

DESCRIPTION
```

Bild 15. Die Logikfunktionen des PAL10L8 in boolschen Gleichungen

der die Belastungen am Kondensator möglichst gering hält.

Man kann leicht sehen, daß für den Kondensator zwei gegensätzliche Bedingungen erfüllt sein müssen. Wenn der Schalter geöffnet ist (Haltezustand), soll der Kondensator die Spannung halten! Dazu ist ein großer Kondensator besonders gut geeignet. Ist der Schalter geschlossen (Folgezustand), soll die Spannung am Kondensator der Eingangsspannung entsprechen, also direkt folgen, was ein kleiner Kondensator am besten kann. Daraus folgt, daß für die Wahl des Kondensators in einem Abtast-Halte-Verstärker ein Kompromiß zwischen diesen beiden Forderungen geschlossen werden muß.

Auf der AD12-Karte werden zwei unterschiedliche Abtast-Halte-Schaltkreise eingesetzt:

Bei dem SMP11 handelt es sich um einen Baustein, der intern mit einer Diodenbrükke beschaltet ist. Durch diese Beschaltung erhält man eine schnelle Schaltung.

Der AD585 ist gemäß dem *Bild 14* aufgebaut, es handelt sich bei dieser Beschaltung um die schnellste Möglichkeit, eine Abtast-Halte-Schaltung zu realisieren. Am Ausgang wurde der Spannungsfolger durch einen Integrator ersetzt, der als Analogspeicher dient.

Hohe Auflösung mit dem A/D-Umsetzer AD574

Der AD574A ist ein kompletter 12-Bit-A/D-Umsetzer für sukzessive Approximation mit Tristate-Puffern für den direkten Anschluß an 8-, 12- oder 16-Bit Mikroprozessoren. Das 12-Bit Ergebnis der Wandlung kann entweder als 12-Bit Datum oder als zwei 8-Bit Daten (einer mit 8-Bit und der andere mit 4-Bit und 4 folgenden Nullen) gelesen werden.

Spannungsversorgung und Layout

Auf die Spannungsversorgung und das Layout für die Schaltung des AD574A muß ein besonderes Augenmerk gerichtet werden. Für das Layout wurde eine Lösung mit weitestgehender Trennung von Digital- und Analogmasse gewählt. So werden die Versorgungsspannungen für den Analogteil über LC-Filter gesiebt und Störsignale werden durch großflächige Masseflächen auf der Bestückungsseite abgeschirmt. Dieser Aufwand ist notwendig, da schon einige Millivolt einen Fehler von einigen Digits verursachen. Im 10-Volt-Eingangsspannungbereich beträgt bei 12 Bit Auflösung das LSB 2,44 mV. Dieser Fehler würde sich besonders bei kleinen Eingangssignalen mit großem Verstärkungsfaktor der Meßverstärker ungünstig bemerkbar machen, so daß sich dieser zusätzliche Aufwand ebenso wie die Blockkondensatoren an den ICs bezahlt macht. Die Filtereingänge können entweder über die Brücken 3A und 4A direkt auf den PC-Bus gelegt oder über die Brücken 3B und 4B mit den externen ±15 V bzw. mit dem internen Spannungswandler verbunden werden. Der interne Spannungswandler wird mit den +12 V des Busses versorgt. Gegen Störspitzen auf der Versorgungsspannung ist ein weiteres LC-Filter vorgesehen. Das LC-Filter sorgt dafür, daß vom Wandler keine Störungen auf den PC-Bus gelangen. Wenn der Umsetzer eingesetzt wird, sind die Spannungen am Eingangsstecker verfügbar.

Auf keinen Fall dürfen bei arbeitendem Wandler an den Eingangsstecker Versorgungsspannungen angeschlossen werden.

Die Brücken für die Spannungsversorgung J3 und J4 haben folgende Bedeutung:

A gesteckt: ±12 V Versorgungsspannung vom PC

B gesteckt: ±15 V Versorgungsspannung vom Umsetzermodul oder extern (Buchsenbelegung beachten)

Der A/D-Wandler hat zwei Analog-Eingänge für 10 Volt und für 20 Volt (Pin 13 und Pin 14), diese Eingänge sind intern mit 5-KΩ-Widerständen als Spannungsteiler beschaltet. Es ist zu beachten, daß das Eingangssignal kleiner als die Versorgungsspannung (0...15 V) sein muß, um eine Zerstörung des teuren Umsetzers zu vermeiden.

Die Steuerlogik

Die Signale CE, $\overline{CS}$ und R/$\overline{C}$ steuern die Operationen des A/D-Umsetzers. Das Signal R/$\overline{C}$ bestimmt, ob Daten eingelesen (R/$\overline{C}$ = 0) oder ausgegeben (R/$\overline{C}$ = 1) werden. Die Register A0 und 12/$\overline{8}$ legen die Länge und das Datenformat fest. Wenn eine Umsetzung mit A0 = 0 gestartet wird, dann handelt es sich um einen 12-Bit-Zyklus, falls während dem Wandlungsstart A0 = 1 ist, erhält man ein 8-Bit-Ergebnis. Während des Lesezyklus gibt A0 an, ob die 8 MSB (A0 = 0) oder die 4 LSB (A0 = 1) ausgegeben werden. Mit der Beschaltung des 12/$\overline{8}$-Eingangs gibt man an, wie die Ausgabe organisiert sein soll:

als zwei 8-Bit Worte (12/$\overline{8}$ auf Digitalmasse) oder als 12-Bit-Wort (12/$\overline{8}$ mit VLogik). Der 12/$\overline{8}$-Eingang ist nicht TTL-kompatibel, so daß er fest verdrahtet werden muß (hier auf Masse). Das Ausgangssignal STS gibt den Status des Umsetzers an. Zu Beginn der Umsetzung wird STS = 1; am Ende der Wandlung wird STS = 0. Außer zur Datenausgabe, kann STS auch zur Steuerung der vorgeschalteten Abtast-Halte-Schaltung verwendet werden.

Tabelle 4 zeigt die Wahrheitstabelle des AD574A. Die Steckbrücken J1, J2 und J7 beeinflussen die Betriebsart des A/D-Umsetzers. Ist die Brücke J1 in Stellung A, wird der A/D-Umsetzer in der unipolaren Be-

Tabelle 4: Wahrheitstabelle des AD574A

CE	$\overline{CS}$	R/$\overline{C}$	12/$\overline{8}$	A0	Operationen
0	X	X	X	X	keine
X	1	X	X	X	keine
1	0	0	X	0	12-Bit-Umsetzung starten
1	0	0	X	1	8-Bit-Umsetzung starten
1	0	1	$+V_{cc}$	X	12-Bit-Ausgabe
1	0	1	Masse	0	8-Bit-MSSB
1	0	1	Masse	1	4-Bit-LSB mit folgenden Nullen

Tabelle 5: Mit den Steckbrücken J1/J2 legt man die Betriebsart des AD574A fest

A gesteckt: 0 V...+10 V Eingangsspannungsbereich B gesteckt: wegen ±15 V Versorgungsspannungsnicht möglich	−5 V... +5 V Eingangsspannungsbereich −10 V...+10 V Eingangsspannungsbereich

```
  Basis-Adresse          schreiben                    lesen
  + Port-Adresse
       0                 8 Bit Wandlung starten       4 Bit LSB lesen
       1                 12 Bit Wandlung starten      8 Bit MSB lesen
       2                 MUX-Kanal selektieren        Statusbyte lesen
       3                 PGA-Verstärkung einstellen   Statusbyte lesen
                           ( wenn verwendet )           ( wie oben )
       4                 Interrupt Flip-Flop löschen

  8-Bit-MSB (Lesen von Basisadresse + 1)

  B7  B6  B5  B4  B3  B2  B1  B0
  D11 D10 D9  D8  D7  D6  D5  D4      bei 12-Bit-Wandlung
  D7  D6  D5  D4  D3  D2  D1  D0      bei  8-Bit-Wandlung

  4-Bit-LSB (Lesen von Basisadresse)

  B7  B6  B5  B4  B3  B2  B1  B0
  0   0   0   0   D3  D2  D1  D0      bei 12-Bit-Wandlung
  0   0   0   0   0   0   0   0       bei  8-Bit-Wandlung

Statusbyte (Lesen Basisadresse + 2 oder 3)

  B7  B6  B5  B4  B3  B2  B1  B0
  :   :   :   S4  S3  S2  S1  S0      selektierter Kanal - MUX
  :   V0  V1                          Verstärkung PGA200
  0                                   Wandlung beendet
  1                                   Wandlung dauert an

Schreibvorgänge:

8-Bit-Wandlung starten (Basisadresse)

  B7  B6  B5  B4  B3  B2  B1  B0
  x   x   x   x   x   x   x   x       Datenbyte beliebig

12-Bit-Wandlung starten (Basisadresse + 1)

  B7  B6  B5  B4  B3  B2  B1  B0
  x   x   x   x   x   x   x   x       Datenbyte beliebig

MUX-Kanal selektieren (Basisadresse +2)

  B7  B6  B5  B4  B3  B2  B1  B0
  x   x   x   S4  S3  S2  S1  S0      selektierter Kanal (AD 7506)
  x   x   x   S4  x   S2  S1  S0      selektierter Kanal (AD 7507)

PGA200-Verstärkung einstellen (Basisadresse + 3)

  B7  B6  B5  B4  B3  B2  B1  B0
  x   x   x   x   x   x   V1  V0
                          0   0       Verstärkung = 1
                          0   1       Verstärkung = 10
                          1   0       Verstärkung = 100
                          1   1       Verstärkung = 1000

Interrupt-Flipflop löschen

  B7  B6  B5  B4  B3  B2  B1  B0
  x   x   x   x   x   x   x   x       Datenbyte beliebig
```

Bild 16. Die E/A-Adressen der A/D-Karte

triebsart betrieben. In Stellung B arbeitet er in der bipolaren Betriebsart. Die Bedeutung der Brücke J2 geht aus *Tabelle 5* hervor. Bei massebezogenen Eingangssignalen muß man die Brücke J7 in Stellung A stecken, jedoch bei differentiellen Eingangssignalen in Stellung B.

Die Schnittstelle zum PC-Bus

Die Decodierung der Karte erfolgt mit IC 7 und IC 9. Bei dem 74LS85 handelt es sich um einen 4-Bit-Vergleicher. Mit der Brücke J6 wird die Basisadresse eingestellt. Wenn die anliegenden Daten an den Eingängen P0...P3 den an den Eingängen Q0...Q3 eingestellten Adressen entsprechen, geht der P = Q – Ausgang auf Eins. IC 7 decodiert die Adressen A3...A6 und IC 9 die Adressen A7...A9, so daß der untere Bereich A0...A2 für sieben Portadressen frei bleibt, wovon fünf genutzt werden.

Die Basisadresse wird mit der Brücke J6 eingestellt (*Tabelle 6* und *Tabelle 7*). Bei der Einstellung der Basisadresse ist auf den freien Port-Adreßraum im PC zu achten, der für Prototypen-Karten im Adreßbereich 0300H...031FH liegt.

Die decodierten Signale werden im PAL10L8 noch mit den Port-Adressen und drei Bus-Signalen verknüpft (*Bild 15*).

Die Verknüpfung des P=Q-Signals mit AEN (Adress Enable) stellt sicher, daß während eines DMA-Zyklus keine E/A-Adresse angesprochen wird.

$\overline{IOW}$ und $\overline{IOR}$ (I/O Write und I/O Read) schalten die Richtung des Datentransfers des bidirektionalen Datentreibers 74LS245 (IC 17) um. Bei Lesezugriffen auf die Karte liegt am DIR-Eingang L-Pegel an, woraufhin der interne Datenbus (AD12 Daten-Bus DO...D7) den Zustand des Datenbusses bestimmt. In allen anderen Fällen wird der DIR-Eingang auf H gehalten, d. h. der Zustand des Rechner-Datenbusses bestimmt den Zustand des interen Datenbusses. Der Datentreiber ist ständig aktiviert, Anschluß 19 (Chip Enable) liegt fest auf Masse. Bei einem Schreibzugriff auf den Latch-Baustein 74LS175 erzeugt das PAL mit dem D-Flipflop 74LS74 (IC 12) einen Impuls zur Übernahme der vier niederwertigen Bits des internen Datenbusses auf die Q-Ausgänge des 74LS175. Der Zustand der Q-Ausgänge bleibt gleich, bis ein anderes Byte in die MUX-Register des AD7506 bzw. AD7507 geschrieben wird. Die Übernahme erfolgt mit einem H-Signal vom 74LS74 (IC 12). Somit liegen die Daten an den Decoder-Eingängen des Analog-Multiplexers an und bestimmen, welcher Eingang mit dem A/D-Umsetzer ver-

Tabelle 6: Einstellung der Basis-Adresse mit Brücke J6

Position	Adreßleitung
A	A9
B	A8
C	A7
D	A6
E	A5
F	A4
G	A3

Tabelle 7: Einstellbeispiele für die Basis-Adressen

Port-Adresse	A A9	B A8	C A7	D A6	E A5	F A4	G A3
300H	0	0	–	–	–	–	–
308H	0	0	–	–	–	–	0
310H	0	0	–	–	–	0	–
318H	0	0	–	–	–	0	0

0: Brücke offen; –: Brücke gesteckt

Tabelle 8: Belegung der Interrupts im PC

Interrupt	Bedeutung
IRQ0	verbunden mit Kanal 0 des Zeitgebers 8253-5
IRQ1	Tastatur
IRQ2	frei für Erweiterungen
IRQ3	frei für Erweiterungen
IRQ4	serielle Schnittstelle
IRQ5	frei für Erweiterungen
IRQ6	Floppy-Controller
IRQ7	paralleler Drucker -Port

bunden ist. Die Eingänge des Analog-Multiplexers sind noch mit dem Datentreiber 74LS245 (IC 16) verbunden, an dem auf dem höchstwertigen Bit der Status des Umsetzers anliegt. Somit kann über diese Adresse (Basisadresse + 2) der Status des AD574A und der ausgewählte Eingang gelesen werden. IC 8 (74LS74) speichert mit einem Signal des PALs den Verstärkungsfaktor des PGA 200. Dieser Zustand kann ebenfalls über IC 16 ausgelesen werden.
Aus dem Statussignal des AD574A wird mit den D-Flipflop 74LS74 ein Interruptsignal erzeugt, das man mit der Brücke J5 auf den Bus legen kann. Dabei ist auf die Belegung der Interrupts zu achten (*Tabelle 8*).
Die Belegung der E/A-Adressen ist in *Bild 16* dargestellt.

Bestückung der Platine

Über dieses Thema wurde in mc schon genug geschrieben, so daß hier nur gesagt sei, daß mit den Fassungen für die ICs – die alle gesockelt sein sollten – begonnen wird, anschließend werden die Widerstände und Kondensatoren eingelötet. Bei den Tantalkondensatoren ist auf die richtige Polung zu achten, der Plusanschluß ist am Kondensator, sowie auf der Platine durch ein + gekennzeichnet. Bei der Inbetriebname der Karte sollte man vorsichtig zu Werke gehen. Die teueren Bausteine sollte man erst dann einsetzen, wenn man sicher ist, daß die benötigten Versorgungsspannungen an den entsprechenden Pins der Fassung anliegen. Aus den *Bildern 17...19* und der *Tabelle 9* kann man alle Angaben zur Herstellung und Bestückung der Platine entnehmen.

Zum Abgleich dienen R3...R6:

R3 dient zum Offsetabgleich beim unipolaren Betrieb, R4 zur Einstellung der Verstärkung, R5 zum Offsetabgleich beim bipolaren Betrieb und R6 wird zum Offsetabgleich der Bausteine AD524 und PGA200 benötigt.

Zum Offsetabgleich mit R6 werden die Eingänge der Multiplexer auf Masse gelegt und der Verstärkungsfaktor auf 1000 gesetzt. Nun muß man mit dem Spindeltrimmer R6 am Ausgang des PGA200 eine Spannung von 0 V (±1 mV) einstellen. Damit ist die Analog-Strecke bis zum Abtast-Halte-Verstärker abgeglichen.

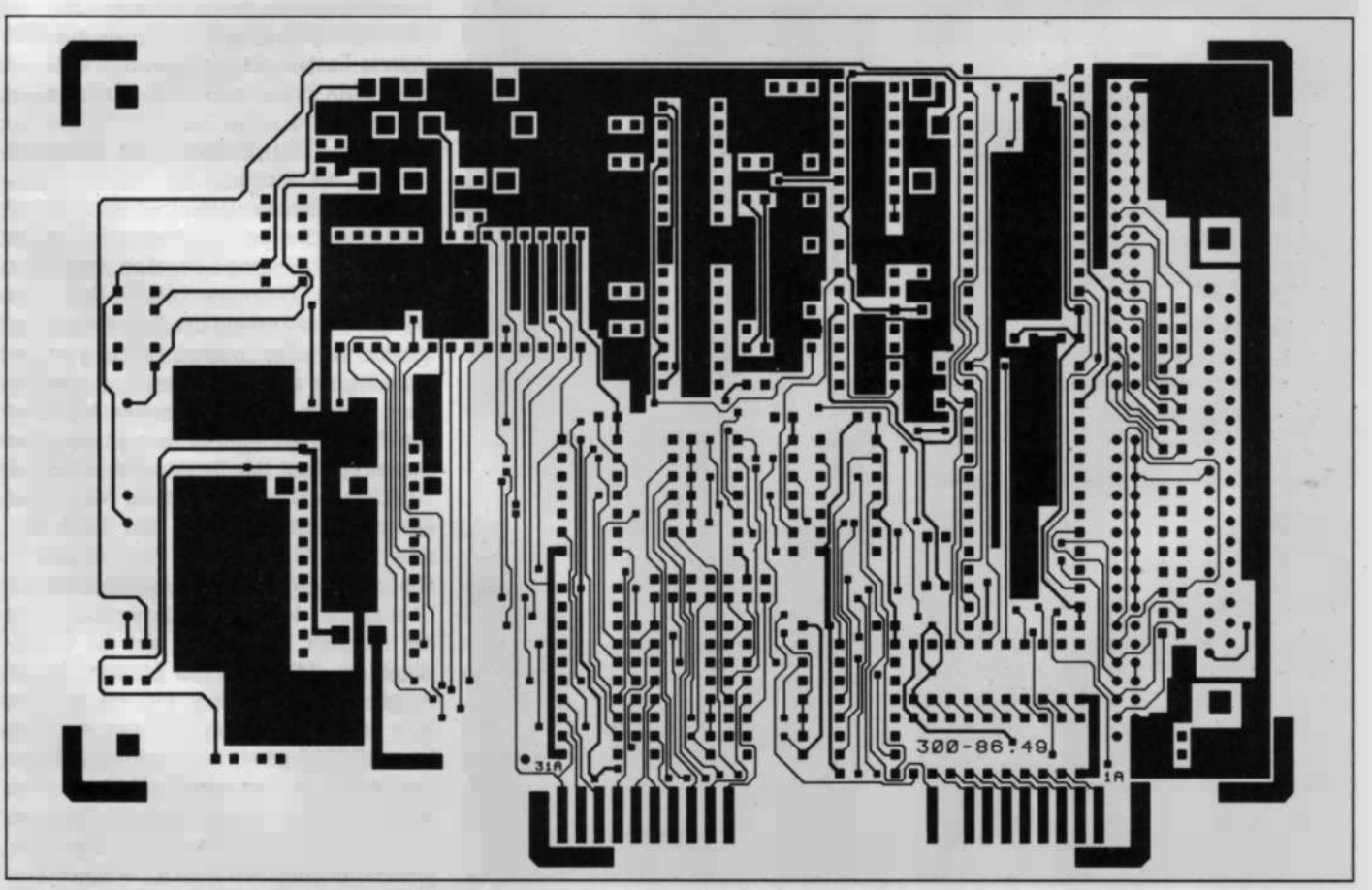


Bild 17. So sieht die Bestückungsseite der Platine aus

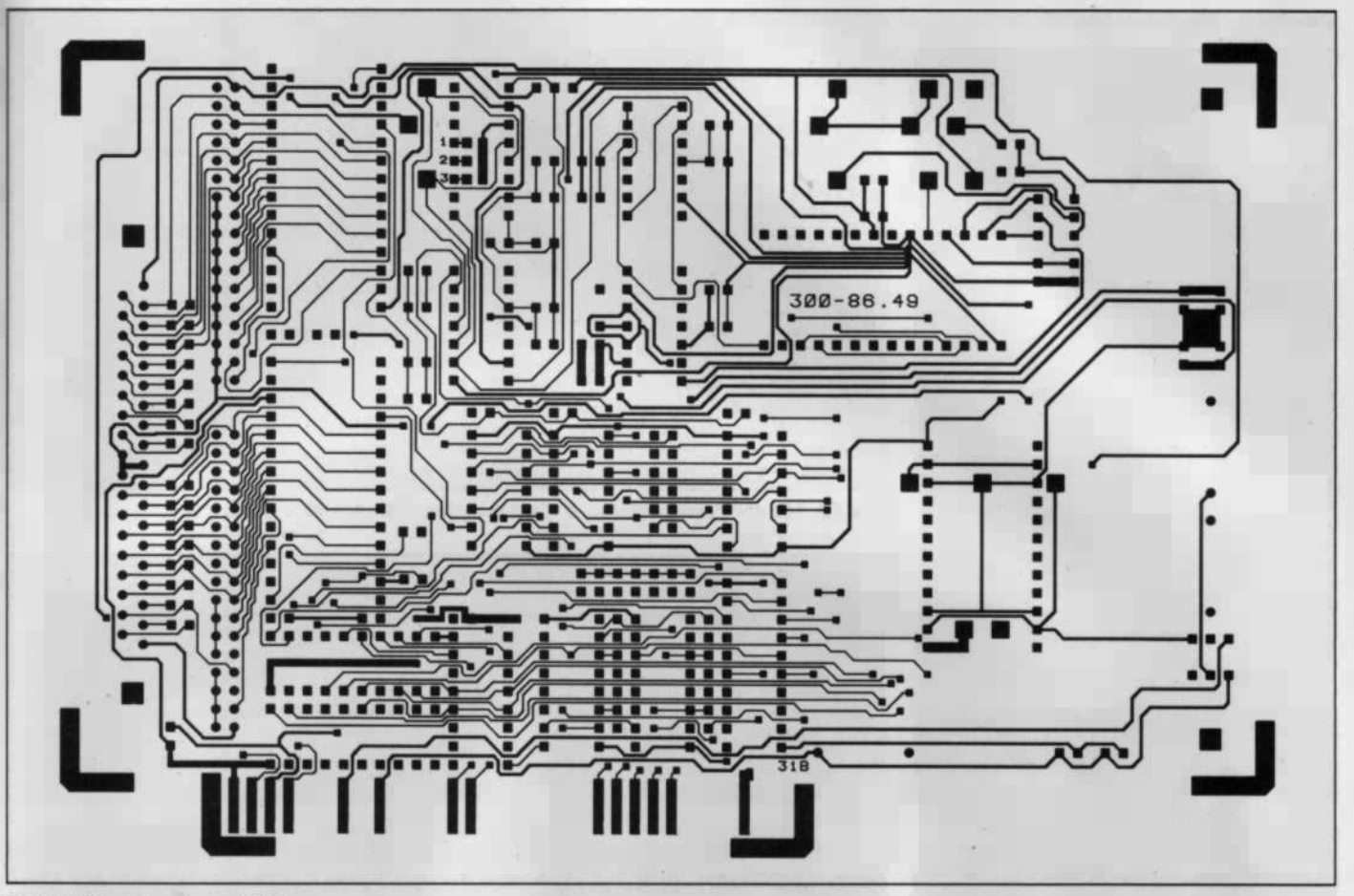


Bild 18. Die Lötseite der Platine

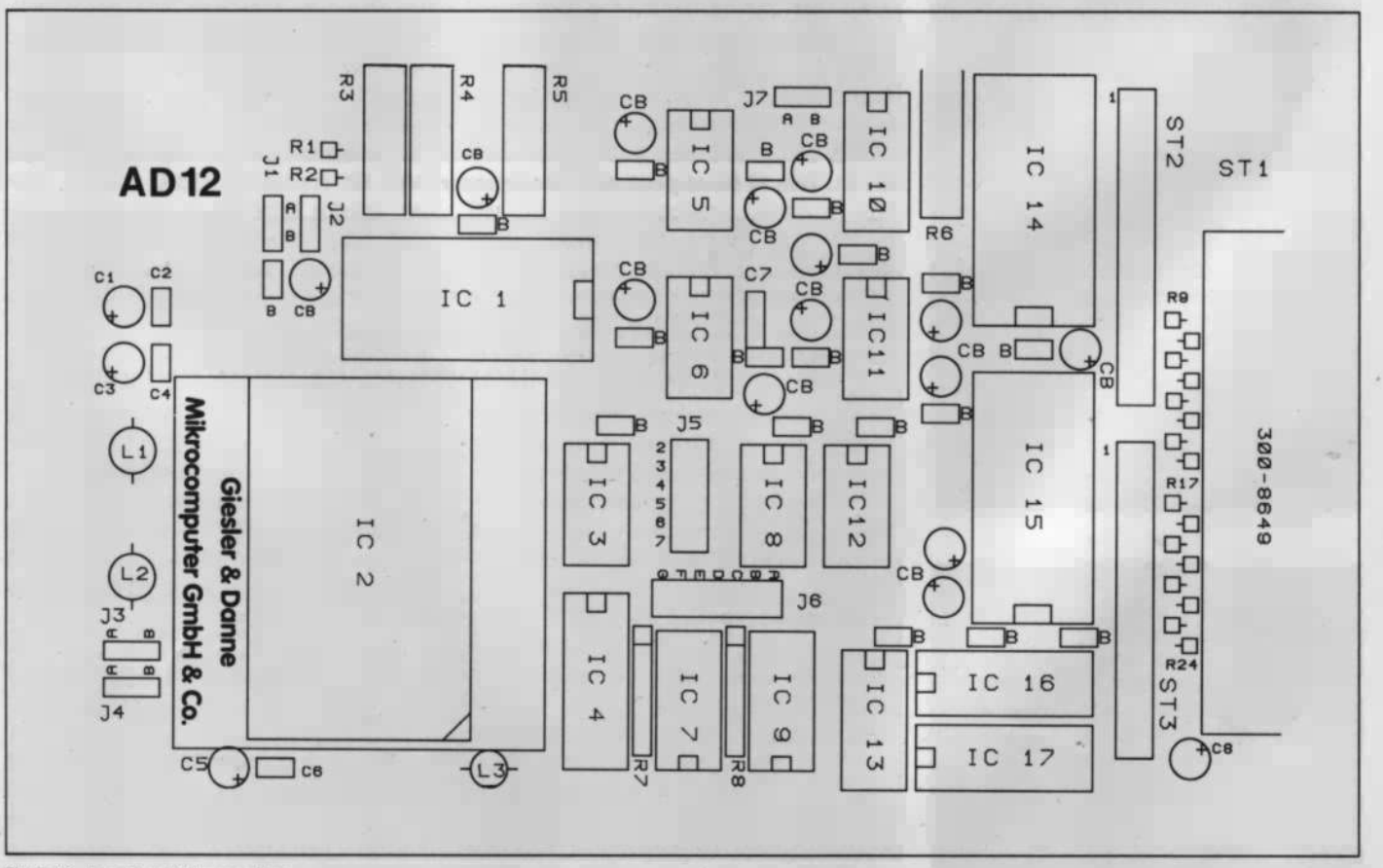


Bild 19. Der Bestückungsplan

Tabelle 9: Die Stückliste

IC 1	AD574	
IC 2	siehe Optionen	
IC 3	7405	
IC 4	PAL1018	
IC 5	AD585	*
IC 6	SMP11	*
IC 7, 9	74LS85	
IC 8, 12	74LS74	
IC 10	AD524	*
IC 11	PGA200	*
IC 13	74LS175	
IC 14, 15	AD7506/07	
IC 16, 17	74LS245	
R 1	100 kΩ 1 %	
R 2	100 Ω 1 %	
R 3	Spindeltrimmer 100 kΩ	
R 4, 5	Spindeltrimmer 100 Ω	
R 6	Spindeltrimmer 10 kΩ/100 kΩ (siehe Schaltplan)	
R 7, 8	Widerstandsnetzwerk 6 × 4,7 kΩ	
R 9...24	500 Ω 1 % (bei AD7507)	
C 1, 3	Tantal 1 µF	
C 2, 4, 6	100 nF	
C 5, 8	Tantal 10 µF	
C 7	3,9 nF	
C 9	10 µF Tantal 14mal, Block	
C 10	10 nF keram. 14mal, Block	
C 11	100 nF keram. 4mal, Block	
L 1, 2	100 µH	
ST1	Buchse Sub-D, 37pol.	

IC-Fassungen

3 Stück	28polig
3Stück	20polig
6Stück	14polig
4 Stück	16polig

Stiftleisten

2 Stück	34polig, zwei Reihen
1 Stück	14polig, zwei Reihen
5 Stück	3polig
1 Stück	12polig, zwei Reihen
13	Steckbrücken

Optionen

IC 2	DC/CD-Wandler
L3	Drossel 100 µH

*: Bestückung entweder mit AD524 oder PGA200 (Meßverstärker) und mit SMP11 oder AD585 (Abtast-Halte-Verstärker). Nicht beide Meßverstärker oder beide Abtast-Halte-Verstärker bestücken (Zerstörungsgefahr)

```
procedure lesen_8;
var Mux  : byte;
    wert : byte;
begin
  ClrScr;
  GotoXY(10,10); write('8 Bit Abgleich ');
  GotoXY(10,12); write('Welcher Multiplexer soll angesteuert werden
0..15 ? ');
  readln(mux);
  port[$302]:= mux;          { Mux selektieren }
  port[$303]:= 0;            { Verstaerkung PGA200 1-fach }
  repeat
    port[$300]:=0;           { Wandlung 8 Bit starten }
    delay(50);               { um die Lesbarkeit zu verbessern }
    repeat
    until port[$302] < 128;    { Statusbyte lesen }
    wert := port[$301];
    GotoXY(10,16); ClrEol;
    write(wert:4,' Digit');     { 8 Bit wert lesen und ausgeben }
    GotoXY(10,20); ClrEol;
    write(wert*0.04:4:2,' Volt');
  until keypressed;
end;
```

Bild 20. Mit der Prozedur lesen_8 führt man den 8-Bit-Abgleich durch

```
procedure lesen_12;
var Mux : byte;
    low : byte;
    high: byte;
begin
  ClrScr;
  GotoXY(10,10); write('12 Bit Abgleich ');
  GotoXY(10,12); write('Welcher Multiplexer soll angesteuert werden
0..15 ? ');
  readln(mux);
  port[$302]:= mux;              { Mux selektieren }
  port[$303]:= 1;                { Verstaerkung PGA200 1-fach }
  repeat
    port[$301]:=0;               { Wandlung 12 Bit starten }
    delay(50);                   { um die Lesbarkeit zu verbessern }
    repeat
    until port[$302] < 128;    { Statusbyte lesen }
    high := port[$301];
    low  := port[$300] and 16;
    GotoXY(10,16); ClrEol;
    write(high*16+low:4,' Digit');     { 12 Bit wert lesen und ausgeben
}
    GotoXY(10,20); ClrEol;
    write((high*16+low)*0.0024:4:2,  Volt');
  until keypressed;
end;
```

Bild 21. Zum 12-Bit-Abgleich dient die Prozedur lesen_12

```
procedure samp_8;
{*************************************************************************}
{  im aufrufenden Program muss 1 array definiert sein :                   }
{  wert    : array[1..Anzahl] of byte                                     }
{  Anzahl muss mit dem Hex - Wert in 0003 uebereinstimmen                 }
{  jetzt 640 dezimal                                                      }
{*************************************************************************}
begin
        inline(
{0000)                          {start:                                   }
{0000)$BF/wert                  {   mov   di, offset wert                 }
{0003)/$BB/$80/$02              {   mov   bx, 640d      ; array groesse   }
{0006)                          {loop:                                    }
{0006)/$BA/$00/$03              {   mov   dx, [lsb]                       }
{0009)/$EE                      {   out   dx, al                          }
{000A)/$BA/$02/$03              {   mov   dx, [stat]                      }
{000D)                          {wait:                                    }
{000D)/$EC                      {   in    al, dx        ; einlesen status}
{000E)/$A8/$80                  {   test  al, 10000000b                   }
                                {                 ; ob wandler aktiv = 128}
{0010)/$75/$FB                  {   jnz   wait ; wenn nicht, dann wait    }
{0012)/$BA/$01/$03              {   mov   dx, [msb]     ;  msb laden      }
{0015)/$EC                      {   in    al, dx        ;  wert einlesen  }
{0016)/$88/$05                  {   mov   [es:di], al   ; al nach wert    }
{0018)/$47                      {   inc   di     ; array + 1 hochzaehlen  }
{0019)/$4B                      {   dec   bx             ; Zaehler - 1    }
{001A)/$75/$EA                  {   jnz   loop                            }
{001C)                          {   cseg  ends                            }
) end;
```

Bild 22. Die Prozedur samp_8 liest das Ergebnis der 8-Bit-Umsetzung aus

```
procedure samp_12;
{************************************************************************}
{  im aufrufenden Program muessen 2 array's definiert sein :             }
{  wert    : array[1..Anzahl] of integer                                 }
{  wandel  : array[1..2*Anzahl] of byte                                  }
{  2*Anzahl muss mit dem Hex - Wert in 0003 uebereinstimmen              }
{  jetzt 1200 dezimal                                                    }
{  Array wert enthaelt Anzahl Werte , jetzt 600                          }
{************************************************************************}
begin
       inline(
{0000)                    {start:                                       }
{0000)$BF/wandel          {     mov   di, offset wandel                 }
{0003)/$BB/$B0/$04        {     mov   bx, 1200d    ; array groesse      }
{0006)                    {loop:                                        }
{0006)/$BA/$01/$03        {     mov   dx, [msb]                         }
{0009)/$EE                {     out   dx, al                            }
{000A)/$BA/$02/$03        {     mov   dx, [stat]                        }
{000D)                    {wait:                                        }
{000D)/$EC                {     in    al, dx      ; einlesen status)    }
{000E)/$A8/$80            {     test  al, 10000000b                     }
                          {                 ; ob wandler aktiv = 128)   }
{0010)/$75/$FB            {     jnz   wait ; wenn nicht, dann wait)     }
{0012)/$BA/$01/$03        {     mov   dx, [msb]         ; msb laden)    }
{0015)/$EC                {     in    al, dx       ;  wert einlesen)    }
{0016)/$88/$05            {     mov   [es:di], al ;  al nach wert )     }
{001B)/$47                {     inc   di    ; array + 1 hochzaehlen)    }
{0019)/$BA/$00/$03        {     mov   dx, [lsb]    ;  lsb laden     )    }
{001C)/$EC                {     in    al, dx       ;  Wert einlesen)    }
{001D)/$88/$05            {     mov   [es:di], al ; al nach wandel)     }
{001F)/$47                {     inc   di    ; array + 1 hochzaehlen)    }
{0020)/$4B                {     dec   bx          ; Zaehler.- 1    )    }
{0021)/$75/$E3            {     jnz   loop                              }
{0023)                    {     cseg  ends                              }
);
  for i:=1 to Anzahl do wert[i] := ((16*wandel[2*i+1]+(Wandel[2*i+2]
and 16)));
end;
```

Bild 23. Die Prozedur samp_12 liest das Ergebnis der 12-Bit-Umsetzung aus

Für den unipolaren Betrieb läßt sich der AD12 wie folgt abgleichen: Mit dem Trimmer R3 wird die untere Grenze des AD574 abgeglichen. Dies geschieht am besten mit dem Testprogramm nach dem oben beschriebenen Offsetabgleich des Meßverstärkers, mit kurzgeschlossenem Eingang der Multiplexer oder einer Eingangsspannung von 1,22 mV (12 Bit Wandlung) am ADU (Analog Digital Umsetzer) bei einer Verstärkung von 1. R3 wird solange verdreht, bis das Ergebnis, das 0000 0000 0000 entspricht, auf dem Bildschirm erscheint. Mit R4 läßt sich der Maximalwert trimmen, bei einem Eingangssignal von 9,9963 Volt (10 Volt Bereich, 12 Bit Wandlung) wird der Trimmer solange verdreht, bis der Sprung von 1111 1111 1110 nach 1111 1111 1111 vollzogen ist.
Der Abgleich für unipolaren Betrieb erfolgt analog zu dem oben beschriebenen, nur daß für die untere Grenze –4,9988 V (±5-V-Bereich , 12-Bit-Wandlung) angelegt werden und für die obere Grenze 4,9988 V.

Turbo-Pascal-Prozeduren zur Inbetriebnahme

Das Einlesen der Werte kann mit Turbo-Pascal erfolgen. Die eigentliche Meßroutine sollte allerdings (um die Geschwindigkeit des Umsetzers auszunutzen) in Assembler geschrieben sein und als Inline-Code in die Turbo-Pascal-Quelldatei eingebunden werden. Bei den in Bildern *20...23* abgedruckten Listings handelt es sich um Turbo-Pascal-Routinen zum Auslesen bei 8-Bit- (Prozedur samp_8), 12-Bit-Umsetzung (Prozedur samp_12) und zum Abgleich. In den Prozeduren zum Abgleich der A/D-Umsetzer gelten die Umrechnungen in Volt bei einer Verstärkung von 1 und einem Eingangsspannungsbereich von 0...10 V.
Das in den Prozeduren samp_8 und samp_12 verwendete Array wird jeweils im Hauptprogramm definiert. Es ist also global, da es sich im gleichen Segment befindet wie die aufgerufene Prozedur. Die Anzahl der Abtastwerte wird über die Konstante Anzahl festgelegt, die auch im Hauptprogramm definiert werden muß. Sie wird allerdings nicht als Konstante übergeben, sondern wird in Zeile 3 des Inline-Codes im Hex-Code eingefügt. Die Notation ist natürlich, wie bei Assembler üblich, erst das High-Byte und dann das Low-Byte. Der dezimale Wert 500 entspricht 01F4H und wird in Zeile 3 wie folgt eingefügt: /$BB/$F4/$01.
Als Basisadresse liegt den Listings die Adresse 300H zugrunde, die gegebenenfalls geändert werden muß. Auch hier gilt erst High-Byte und dann Low-Byte.
Vor der eigentlichen Meßroutine muß noch der Multiplexer ausgesucht werden, an dem das Eingangssignal anliegt. Ebenso muß bei Bestückung mit dem PGA200 die Verstärkung festgelegt werden. Die Zeile für den Multiplexer lautet in Turbo-Pascal: port[$302] := Nummer des MUX.
Der Verstärkungsfaktor wird entsprechend ausgewählt.
Es ist natürlich auch denkbar, die Meßroutinen für die Meßwerterfassung auf mehrere Eingangssignale zu erweitern, wobei die Umschaltung der Multiplexer dann in den Inline-Code eingefügt werden muß. Ist die Karte mit dem PGA200 bestückt, kann dann für jeden Kanal beim Umschalten eine andere Verstärkung gewählt werden und somit der Bereich des Wandlers voll ausgenutzt werden.
Wem die Herstellung der A/D-Karte zu kompliziert ist, der kann sie beim Elektronikladen Detmold als Bausatz oder als Fertigkarte beziehen.

Literatur

[1] *Tietze U., Schenk Ch.:* Halbleiter-Schaltungstechnik, Springer-Verlag, 1985
[2] Datenblatt AD574A, Analog Devices
[3] Datenblatt AD885, Analog Devices
[4] Datenblatt SMP11, Analog Devices
[5] Datenblatt AD524, Analog Devices
[6] Datenblatt PGA200, Analog Devices
[7] Datenblatt AD7506/07, Analog Devices
[8] *Strauß, Dieter:* Der gläserne PC-Bus, in: Das PC-Sonderheft, Seite 52, Franzis Verlag, München

Gerd Graf

Der mc-modular-AT

Teil 8: MS-DOS für Anfänger

Viele Computer-Anfänger kennen nicht die wichtigsten MS-DOS-Befehle und wissen auch nicht, wie man eine Festplatte einrichtet. Wir erklären die meist gebrauchten MS-DOS-Befehle, die natürlich nicht nur auf dem mc-modular-AT, sondern auf jedem MS-DOS-Rechner laufen.

Die Grundlage der Kommunikation Mensch-Computer bildet das Betriebssystem; beim mc-modular-AT wird MS-DOS V3.21 oder V3.3 verwendet. MS steht für Microsoft, ein Softwarehaus aus den USA, das dieses Betriebssystem entwikkelt hat, und DOS steht für „Disk Operating System", was soviel wie Disketten-(Platten-)Betriebssystem bedeutet.
Endlose Seiten Papier sind in Artikel- und Buchform bereits über MS-DOS bedruckt worden. Dieser Beitrag beschäftigt sich vorwiegend mit dem Einstieg in das Betriebssystem und mit den Schwierigkeiten, die ein Anfänger haben kann. Es gibt über MS-DOS hervorragende Literatur [1, 2, 3], auf die wir hier hinweisen möchten.

Die ersten Gehversuche

Für jeden, der schon einmal mit einem Computer gearbeitet hat, ist das Thema Backup oder Datensicherung wichtig.
Arbeiten Sie nie mit der Original-Diskette des Betriebssystems. Dieses Original ist für Ihren Wohnzimmerschrank, Ihren Safe oder einen ähnlich sicheren Aufbewahrungsort geeignet.
Bevor man mit dem Arbeiten beginnt, sollte man sich zuerst eine Kopie der Betriebssystemdiskette herstellen.
Die Katze beißt sich hier in den Schwanz, denn um das Betriebssystem kopieren zu können, muß man es erst einmal starten. Aber bitte nur das erste Mal von den Original-Disketten starten -danach muß man sofort eine Sicherungskopie erstellen. Sicher ist sicher!

MS-DOS von Diskette booten

MS-DOS wird auf zwei Disketten geliefert. Zum Booten des Systems verwendet man die erste Diskette – sie ist als solche gekennzeichnet – steckt diese bei ausgeschaltetem Computer in das Laufwerk A (meist das obere oder das einzige 5¼-Zoll-Laufwerk), schließt die Laufwerksklappe und schaltet den Computer ein.
MS-DOS meldet sich nach einiger Zeit, und

```
A>KEYBGR

A>DATE
Aktuelles Datum Don  7.01.1988
Neues Datum: (TT-MM-JJ):

A>TIME
Uhrzeit: 21.26.51,56
Neue Uhrzeit:

A>VER

MS-DOS Version 3.21

A>
A>
```

Bild 1. So meldet sich MS-DOS nach dem Booten

auf dem Bildschirm erscheinen die oberen zwei Zeilen von *Bild 1*.
Bei einem Rechner mit Echtzeituhr wie z. B. dem mc-modular-AT wird das aktuelle Datum angezeigt. Man kann jedoch auch ein anderes Datum eingeben.
Sofern man mit dem Setup-Programm die eingebaute Uhr richtig gestellt hat, müßte nun das richtige Datum erscheinen. Später werden wir eine Möglichkeit kennenlernen, die immer wiederkehrende Datumseingabe zu unterdrücken, da der mc-modular-AT über eine gepufferte Echtzeituhr verfügt. Die Datumseingabe stammt noch aus Zeiten, in denen zum Standlieferumfang noch nicht die Echtzeituhr gehörte.
Nachdem man das Datum mit dem Drücken der Eingabetaste bestätigt hat, gibt MS-DOS die aktuelle Uhrzeit an, die man ebenfalls mit der Eingabetaste bestätigen oder gegebenenfalls neu eingeben kann (Format beachten: HH:MM für Stunde:Minute). Danach wird durch den Aufruf VER die aktuelle Versionsnummer von MS-DOS angegeben und mit dem Prompt A> zeigt MS-DOS, daß es nun bereit zur Entgegennahme von Befehlen ist. Bevor man weiterarbeitet, sollte man eine Sicherungskopie der Quelldiskette anfertigen.

Sicherheit durch Kopieren

Um schnell eine 1:1-Kopie einer Diskette, in unserem Fall der Disketten des Betriebssystems, herzustellen, verwendet man den Befehl „DISKCOPY".
Seine Syntax lautet:

DISKCOPY [Quellaufwerk:] [Ziellaufwerk:]

Dieser Befehl kopiert den gesamten Inhalt der Diskette im Quellaufwerk auf die Diskette im Ziellaufwerk, wobei die Diskette im Ziellaufwerk nicht einmal formatiert sein muß. Alle Daten, die sich auf der Zieldiskette befinden, werden überschrieben.
Hierbei verbirgt sich bei DISKCOPY eine ganz gemeine Falle:
Vertauscht man – vielleicht spät am Abend, oder in der Aufregung – die Laufwerksbezeichnungen für Quell- und Ziellaufwerk, kopiert DISKCOPY gnadenlos den vielleicht sinnlosen Inhalt der Zieldiskette auf die Quelldiskette.
Profis schützen sich, indem sie vor dem Kopieren die Quelldiskette mit einem Schreibschutzaufkleber versehen. Dadurch kann dann – auch bei Fehlbedienung – die wertvolle Quelldiskette nicht überschrieben werden.
Wichtig ist, daß unter MS-DOS das Ziel- und das Quellaufwerk auch identisch sein können. Dadurch ist es möglich, ohne große Klimmzüge (wie z. B. bei CP/M 2.2) mit nur einem Diskettenlaufwerk komplette Disketten zu kopieren. Man gibt einfach ein: DISKCOPY A: A:. Vom System wird man darauf aufgefordet, zunächst die Quelldiskette und anschließend die Zieldiskette einzulegen.

Hierbei gilt das oben gesagte über den Schreibschutz, da ein Vertauschen der Disketten auch hier leicht möglich ist. Mit DISKCOPY kann man Disketten jeweils nur im gleichem Format kopieren.

Disketten formatieren

Jede neue Diskette muß, bevor mit ihr gearbeitet werden kann, formatiert werden. Darunter versteht man das Aufbringen von Spur- und Sektorinformation auf die noch leere Diskette. Beim Formatieren werden alle Daten, die bisher auf der Diskette waren, unwiederbringlich gelöscht. Auch alle bekannten Hilfsprogramme schaffen es nicht, Daten auf einer neuformatierten Diskette wieder herzustellen.
Profis sind deshalb wieder etwas großzügiger im Kauf von Disketten und verwenden zum Formatieren nur neue Disketten – das versehentliche Überschreiben eines wichtigen Programmes oder sonstiger wichtiger Daten, ist wesentlich schmerzlicher als der Kauf einiger, doch mittlerweile recht preiswerter Diskettenpackungen.
Um neue Disketten zu formatiern, verwendet man den Befehl:
FORMAT Laufwerk: [/1][/4][/8][/N:XX] [/T:XX][/V][/S]
Die in eckigen Klammern angegebenen Parameter steuern den Ablauf der Formatierung, werden sie weggelassen, verwendet MS-DOS Standardvorgaben.
/1 bedeutet, daß nur eine Seite der Diskette formatiert werden soll; mit /4 kann man mit einem AT-Laufwerk 360K-Disketten erzeugen, die allerdings etwas mit Vorsicht zu genießen sind. /8, /N und /T sind für Sonderfälle vorgesehen:
/8 formatiert 8 (statt 9 oder ...) Sektoren pro Spur (wird eigentlich nie benötigt); mit /N:XX gibt man eine freie Anzahl von Sektoren pro Spur, mit /T:XX eine Anzahl von Spuren je Diskette an.
Der Parameter /V ist wichtig: Er erlaubt es, nach dem Formatieren eine Diskettenbezeichnung anzugeben, einen Namen, der bis zu 11 Zeichen lang sein kann.
Gibt man /S ein, werden die versteckten Dateien des Betriebssystems kopiert und man kann die so erzeugte Diskette später als Systemdiskette verwenden.
Um eine neue Diskette im mc-modular-AT im Laufwerk A mit 1,2 MByte zu formatieren, gibt man ein:
FORMAT A: /V
Achtung: Der Format-Befehl funktioniert auch bei Festplatten!
Dies kann fatale Folgen haben, zumal die Festplatte keinen Schreibschutz-Aufkleber hat. Bitte also beim FORMAT-Befehl Vorsicht walten lassen!

Größe (Zoll)	Beschreibung	Kapazität (Byte)	Spuren/ Seite	Sektoren / Spur	
5,25	Einseitig	160 K	40	8	(selten!)
5,25	Einseitig	180 K	40	9	
5,25	Doppelseitig	320 K	40	8	(selten!)
5,25	Doppelseitig	360 K	40	9	
5,25	Hohe Kapazität (AT)	1,2 M	80	15	AT
3,25	Doppelseitig	720	80	9	PS/2 ab /30
3,25	Doppelseitig	1,4 M	80	18	PS/2 ab /50

Alle Formate: 512 Byte pro Sektor

Bild 2. Gebräuchliche Disketten-Formate

Bild 2 zeigt die möglichen Disketten-Formate unter MS-DOS, wobei die 160-KByte- bzw. 320-KByte-Versionen kaum verwendet werden. Das Standard-Austauschformat ist heute noch 360 KByte/5¼-Zoll – bald wird es wohl 3½-Zoll/720 KByte sein.
Das Standard-Format eines AT – und natürlich auch des mc-modular-AT – ist 1,2 MByte/5¼-Zoll.
Ein solches Laufwerk gehört zum Standard-Lieferumfang des mc-modular-AT. Mit diesem Laufwerk ist es möglich, sowohl die 320-/360-KByte-Disketten als auch die 1,2-MByte-Disketten zu lesen und zu beschreiben.
In der Praxis sollte allerdings immer nur der Weg gewählt werden, daß mit diesem Laufwerk 360-KByte-Disketten nur gelesen werden, da der Aufbau der Spurlage bei 1,2-MByte-Disketten ein anderer ist als beim 360-KByte-Laufwerk. Sonst kann es vorkommen, daß mit einem 1,2-MByte-Laufwerk beschriebene ehemalige 360-KByte-Disketten später nicht mehr mit einem 360-KByte-Laufwerk gelesen werden können.
Weitere gebräuchliche 3½-Zoll-Formate (IBM PS/2) sind 720-KByte-Laufwerke sowie 1,44-MByte-Laufwerke.
Mit den 1,44-MByte-Laufwerken können sowohl die 720-KByte-Disketten als auch die 1,44-MByte-Disketten verarbeitet werden.

Dateiname und Baumstruktur

MS-DOS verwendet – wie andere Betriebssysteme – Dateinamen, die aus einem Haupt-Dateinamen (8 Zeichen) und der Dateinamenerweiterung (3 Zeichen) bestehen. Dateiname und -erweiterung werden durch einen Punkt getrennt. Dieser Punkt erscheint jedoch später nicht auf dem Bildschirm. Die Erweiterung dient dazu, Dateien grob zu klassifizieren. So erhalten ablauffähige Programme die Erweiterung EXE (für Executable) oder COM (stammmt noch aus der CP/M-Zeit). Weitere gängige Erweiterungen sind SYS (für Systemdateien), BAS (für Basic-Dateien), TXT (für Textdateien) und BAT (für Stapeldateien, deren Funktion wir später beschreiben).
Das Inhaltsverzeichnis einer Diskette kann man sich mit dem Befehl DIR anschauen. *Bild 3* zeigt das Directory der Betriebssystemdisketten 1 und 2.
Sobald auf einer Diskette oder der Festplatte eine größere Anzahl Dateien sind, erkennt man kaum noch die schnell nach oben gescrollten Dateinamen. Abhilfe schafft der Parameter /P des DIR-Befehls. Wenn man DIR/P eingibt, hält das System die Ausgabe immer nach einem vollen Bildschirm an. Man hat jetzt genug Zeit, den Bildschirm zu lesen. Weiter geht's mit Tastendruck.
Ein wichtiger Punkt beim Betriebssystem MS-DOS ist ebenfalls im Bild 3 erkenntlich. Neben dem Dateinamen wird auch die Größe der Datei in Byte angegeben sowie das Datum und die Zeit, wann diese Datei zum letztem Mal schreibend verändert wurde. Dies ist zur Dokumentation und besonders für BACKUP-Zwecke ein sehr nützlicher Hinweis. Deshalb sollte man darauf achten, daß die im Rechner eingebaute Uhr – sofern vorhanden – auch das tatsächliche Datum anzeigt.
Eine weitere Option der DIR-Befehls ist der Schalter /W. „W" steht für „wide" (breite Anzeige); es werden lediglich die Dateinamen und keine weiteren Information angezeigt. Hierbei erscheinen pro Zeile maximal fünf Dateinamen auf dem Bildschirm.
Oftmals ist es sinnvoll, das Dateiinhaltsverzeichnis auch auf einem Drucker auszugeben, um es z. B. zur Dokumentation zu verwenden. Hierzu kann man die Umleitung der Ausgabe unter MS-DOS verwenden, die mit dem Zeichen „>" eingeleitet wird. Dabei wird die Ausgabe nicht mehr auf den Bildschirm, sondern auf eine Datei oder ein physikalisches Gerät ausgegeben. Mit dem Befehl DIR > INH.TXT wird das

Directory in eine sequentielle ASCII-Datei namens INH.TXT ausgegeben, die z. B. später mit einem Texteditor weiter verarbeitet werden kann. Mit der Umleitung DIR > PRN wird das Directory auf einem angeschlossenen und betriebsbereiten Drucker ausgegeben.

MS-DOS prüft das Vorhandensein dieses Druckers nicht. Ist der Drucker nicht angeschlossen oder nicht bereit, stellt sich das System tot und ist nur durch das gleichzeitige Drücken von Strg+Alt+End (Ctrl-Alt-Del) oder durch Einschalten des Druckers wieder zum Leben zu erwecken.

Die Baumstruktur

Einer der wesentlichen Vorteile des Betriebssystems ist die Möglichkeit „hierarchische" Inhaltsverzeichnisse aufzubauen. Dies ist weit weniger schwierig, als es sich anhört. Man vergleicht ein solches Inhaltsverzeichnis auch mit der Struktur eines Baumes.

Ein anderer Vergleich ist die Archivierung von schriftlichen Unterlagen. Hierzu werden Ordner verwendet, diese mit Registerblätter unterteilt und vielleicht die so geschaffenen Unterregister nochmals unterteilt.

Dies ist unter MS-DOS ebenfalls möglich. Man kann dazu sogenannte Unterverzeichnisse (Subdirectories) anlegen, die sich am besten mit den Registern eines Ordners vergleichen lassen.

Der Befehl, um ein solches Unterverzeichnis anzulegen, lautet unter MS-DOS:

MD [Laufwerk:] name.

MD steht hier für „Make Directory". Auf der Betriebssystemdiskette bzw. auf einer neuen formatierten Diskette ist noch kein solches Unterverzeichnis angelegt.

Um beim Vergleich des Baumes zu bleiben, befindet man sich sozusagen im dicken Stamm. Das Stamm-Directory, auch Wurzelverzeichnis oder Root genannt, wird durch den Suchpfad „\" gekennzeichnet. Mit Hilfe des Befehls „MD" wollen wir nun einen „Ast", also ein neues Unterverzeichnis, erzeugen. Es soll dies der Ast sein, unter dem wir z. B. unsere Basic-Programme ablegen. Der Befehl dazu lautet:

MD BASIC

Unser Baum hat nun einen Ast; bzw. unser Computer ein neues Unterverzeichnis namens BASIC. In dieses Unterverzeichnis gelangt man mit dem MS-DOS-Befehl CD, der steht für „Change Directory". Danach muß man den Namen des Unterverzeichnisses eingeben, in unserem Beispiel CD BASIC.

Nachdem man mit CD BASIC in dieses Unterverzeichnis gelangt ist, sieht man mit DIR den Inhalt dieses neuen Unterverzeichnisses (*Bild 4*). Es erscheinen nun nicht mehr alle Dateien, die z. B. im Stamm vorhanden sind, sondern nur die Dateien innerhalb dieses Unterverzeichnisses.

Dieses Unterverzeichnis ist leer. Doch halt, es enthält zwei Dateien, die ganz seltsam aussehen, sie haben einen Dateinamen"." sowie einen Dateinamen"..".

Es sind dies keine Dateien, sondern wiederum Verweise auf Directories.

Das Directory mit einen Punkt steht für das aktuelle Verzeichnis – in dem wir uns gerade befinden; das Directory mit zwei Punkten steht für das Übergeordnete, also in unserem Beispiel das Stammverzeichnis. Dies benötigen wir, um schnell wieder zurückzukommen. Mit dem Befehl CD.. gelangt man wieder in das Stammdirectory zurück. Tut man das und schaut sich einmal mit DIR das Stammdirectory an, sieht man, daß sich am Ende eine weitere Datei eingeschlichen hat, die ganz seltsam aussieht. Sie hat den Dateinamen BASIC, keine Erweiterung und statt einer Größenangabe die Angabe <DIR> (*Bild 5*). Dies zeigt, daß im Hauptverzeichnis ein Unterdirectory angelegt ist, das eben BASIC heißt.

Dieses Spiel kann man nun fast beliebig tief weitertreiben. In-

```
 Diskette/Platte, Laufwerk A:, hat keinen
 Namen
Verzeichnis von A:\

ANSI     SYS      1651   11.05.87   12.00
APPEND   COM      1734   11.05.87   12.00
ASSIGN   COM      1526   11.05.87   12.00
ATTRIB   EXE      8927   11.05.87   12.00
CHKDSK   EXE     10368   11.05.87   12.00
COMMAND  COM     24412   11.05.87   12.00
DISKCOMP EXE      3984   11.05.87   12.00
DISKCOPY EXE      4992   11.05.87   12.00
DRIVER   SYS      1109   11.05.87   12.00
EDLIN    EXE      7708   11.05.87   12.00
EXE2BIN  EXE      3168   11.05.87   12.00
FC       EXE     15241   11.05.87   12.00
FDISK    EXE     18075   11.05.87   12.00
FIND     EXE      6435   11.05.87   12.00
GRAFTABL EXE      8640   11.05.87   12.00
GRAPHICS EXE     13332   11.05.87   12.00
FORMAT   EXE     11517   11.05.87   12.00
JOIN     EXE      9737   11.05.87   12.00
KEYBDV   EXE      3509   11.05.87   12.00
KEYBFR   EXE      3607   11.05.87   12.00
KEYBGR   EXE      3599   11.05.87   12.00
KEYBIT   EXE      3525   11.05.87   12.00
KEYBSP   EXE      3632   11.05.87   12.00
KEYBSUI  EXE      4945   11.05.87   12.00
KEYBUK   EXE      3513   11.05.87   12.00
LABEL    EXE      2852   11.05.87   12.00
MODE     EXE     14749   11.05.87   12.00
MORE     COM       283   11.05.87   12.00
PRINT    EXE      9048   11.05.87   12.00
RAMDRIVE SYS      6548   11.05.87   12.00
RECOVER  EXE      4353   11.05.87   12.00
REPLACE  EXE      5074   11.05.87   12.00
SORT     EXE      1658   11.05.87   12.00
SUBST    EXE     10623   11.05.87   12.00
SYS      COM      4735   11.05.87   12.00
INFO     TXT       675   11.05.87   12.00
CONFIG   SYS        16   30.10.87   10.14
AUTOEXEC BAT        26   30.10.87   10.15
       38 Datei(en)      58368 Byte frei
 Diskette/Platte, Laufwerk A:, hat keinen
 Namen
Verzeichnis von A:\

BACKUP   EXE     24781   11.05.87   12.00
DEBUG    EXE     15766   11.05.87   12.00
GWBASIC  EXE     80848   11.05.87   12.00
RESTORE  EXE     22921   11.05.87   12.00
SHARE    EXE      8624   11.05.87   12.00
TREE     EXE      9613   11.05.87   12.00
XCOPY    EXE      5648   11.05.87   12.00
        7 Datei(en)     190464 Byte frei
```

Bild 3. Der Inhalt der beiden MS-DOS-Disketten

```
 Diskette/Platte, Laufwerk C:, hat den
 Namen DISK1_VOL1
Verzeichnis von C:\TOOLS\BASIC

.            <DIR>       7.01.88   16.43
..           <DIR>       7.01.88   16.43
       2 Datei(en)  16664576 Byte frei
```

Bild 4. Das Unterverzeichnis BASIC ist noch leer

```
LASIBM   TXT       512   7.01.88   20.16

BILD1    ~XT      1024   7.01.88   15.43

BILD1    TXT      1024   7.01.88   15.55

BASIC         <DIR>       7.01.88   16.43

      92 Datei(en)  16664576 Byte frei
C:\TOOLS>
```

Bild 5. Das Unterverzeichnis BASIC ist angelegt

nerhalb des Directory BASIC kann man mit dem Befehl MD wieder ein Unterdirectory aufbauen usw.
Bei dem Anlegen von Unterverzeichnissen sollte man wie mit den Registern bei den Ordnern verfahren. Macht man es zu kompliziert, hat man nur noch Register und keine beschriebenen Blätter mehr dazwischen.
Mit dem CD-Befehl kann man sich innerhalb der Directories bewegen.
Vom Hauptdirectory kommt man mit CD BASIC in das Unterdirectory BASIC und mit einen weiteren CD-Befehl z. B. in das Unterdirectory „DATEN". Mit dem Befehl CD.. kommt man jeweils eine Ebene höher. Um beim Baumbeispiel zu bleiben, man „klettert" vom Zweig zum Ast und anschließend vom Ast zum Stamm.
Sobald man sich bis zum Stamm zurückgearbeitet hat, wird ein weiterer CD..-Befehl zur Meldung „Ungültiges Verzeichnis von MS-DOS" führen.
Eine typische Directory-Struktur ist in *Bild 6* zu sehen; dieses Bild wurde mit dem NORTON-Hilfsprogramm erzeugt.
Um von einem beliebigen Directory (oder auch von einem anderen Laufwerk) auf eine beliebige Datei zugreifen zu können, muß der gesamte Weg vom Stamm bis zu dieser Datei, der „Suchweg", angegeben werden. Beispiel: Das Betriebssystem meldet sich gerade mit A>, und man möchte eine Datei TEST.DOC im Unterverzeichnis BASIC ansehen:
TYPE C:\BASIC\TEST.DOC
Durch den „Backslash" werden die Bezeichnungen der Directories voneinander getrennt.
Die Syntax ist besonders für den COPY-

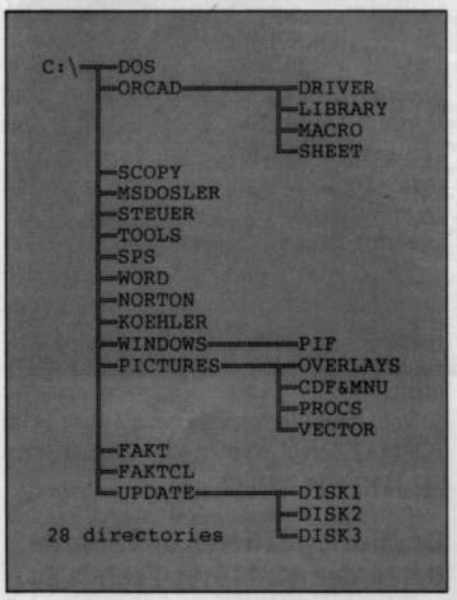


Bild 6. Die Norton-Utilities zeigen die Directory-Struktur

Befehl wichtig, der es erlaubt, von beliebigen Directories zu beliebigen Directories zu kopieren. Die Syntax lautet:
COPY quelle ziel
Quelle und Ziel bestehen aus [Laufwerk:] Suchname.
Der folgende Befehl kopiert die eben erwähnte Datei auf das Laufwerk A: und dort in das Stamm-Directory:
COPY C:\BASIC\TEST.DOC A:
Als Quell-Laufwerk verwendet MS-DOS das gerade aktuelle; ist das Ziel-Laufwerk identisch, muß es nicht explizit angegeben werden. Besonders für die mit MS-DOS unerfahrenen Benutzer ist dieses Verfahren gewöhnungsbedürftig, die Vorteile sind aber – besonders bei vielen Dateien – ganz klar über den Einführungsschwierigkeiten zu sehen.
Sollte man sich beim Ausprobieren dieser Befehle total verrennen, kann man mit dem Betriebssystembefehl PATH feststellen, in welchem Unterverzeichnisse man sich befindet.
Falls man es immer wissen will, gibt man PROMPT $P ein, und der gesamte Suchpfad erscheint. Dieser Befehl eignet sich sehr gut zum Einbinden in die AUTOEXEC.BAT-Datei, mit der wir uns als nächstes beschäftigen.
Mit dem Befehl CD\ kommt man immer ins Stammverzeichnis zurück.
Sofern Sie Ihre Festplatte nicht schon eingerichtet haben, können Sie zunächst mit nur einem Diskettenlaufwerk und MS-DOS arbeiten; geben Sie dazu im SETUP-Menü [4] als Anzahl der Festplatte 0 ein.

Stapelverarbeitung durch Batch-Dateien

Der Begriff Stapelverarbeitung kommt aus grauer Vorzeit der EDV – hier wurden immer wiederkehrende Anweisungen an den Rechner durch Stapel von Lochkarten festgelegt; der Stapel mußte dann jeweils nur einmal mit dem Leser eingelesen werden und der Rechner konnte sequentiell seine Arbeit erledigen.
Genauso funktioniert es unter MS DOS, nur sind die Lochkarten durch eine normale ASCII-Datei mit der Erweiterung .BAT (für Batch-File) ersetzt worden.
MS-DOS sucht nach dem Start nach einer Batch-Datei mit dem Namen AUTOEXEC .BAT und führt die darin enthaltenen Befehle und Kommandos aus.
Die Kommandos in Bild 1 (KEYBGR, DATE, TIME, VER) sind in einer AUTOEXEC.BAT-Datei enthalten.
Mit einem beliebigen Texteditor, der ASCII-Dateien erzeugen kann, können Sie diese Datei ändern oder erweitern.
Sofern Ihr Rechner wie der mc-modular-AT über eine Echtzeituhr verfügt, können Sie die Anweisungen DATE sowie TIME aus der AUTOEXEC.BAT-Datei entfernen. Sie müssen dann nicht mehr nach jedem Start des Betriebssystems diese beiden Parameter eingeben.
Sinnvoll können Sie den oben erwähnten PROMPT-Befehl in die AUTOEXEC mit einbinden; daduch wird der aktuelle Pfad immer angezeigt.
Mit der Anweisung PROMPT PG erhalten Sie nicht nur den aktuellen Pfad, sondern wieder die gewohnte Eingabeaufforderung „>".
Den Befehl KEYBGR sollte man in der AUTOEXEC-Datei stehen lassen, da er den deutschen Tastaturteiber lädt.
Mit in die AUTOEXEC-Datei gehört – sofern Sie mit Hilfe von Unterverzeichnissen etwas Ordnung innerhalb Ihrer Festplatte geschaffen haben – der PATH-Befehl.
Mit Hilfe dieses Befehles teilt man MS-DOS mit, wo es noch nach ablauffähigen Programmen suchen soll, sofern diese im aktuellen Verzeichnis nicht vorhanden sind.
Beispiel: Sie legen für Ihren Texteditor ein eigenes Unterverzeichnis an, möchten jedoch innerhalb eines anderen Unterverzeichnisses mit dem Texteditor arbeiten. Sie müssen nun nicht den Texteditor in dieses aktuelle Unterverzeichnis kopieren, sondern Sie geben einmal innerhalb der AUTOEXEC-Datei mit dem PATH-Befehles an, wo MS-DOS überall suchen muß.
Leider hat der PATH-Befehl einen Haken: er kann nur BAT, EXE- und COM-Dateien in anderen Verzeichnissen finden, nicht jedoch die oft damit verbundenen Overlay-Dateien. Die kann MS-DOS jedoch mit dem APPEND-Befehl finden, der sozusagen den PATH-Befehl für Overlay- und Textdateien darstellt.
Am Ende einer solchen Stapelverarbeitung kann man den Namen eines Programms angeben, das nach dem Start des Computers aufgerufen werden soll. In Batch-Dateien darf man mit IF-Abfragen und Parametern arbeiten. Dies ist in [2] unter Kapitel 4 ausreichend erklärt.

Textverarbeitung mit EDLIN

Mit dem Betriebssystem MS-DOS wird ein sehr einfacher Texteditor EDLIN geliefert. Es scheint Tradition bei den Betriebssystemen zu sein, daß die Qualität des Texteditors in keinem Verhältnis zur Qualität des Betriebssystems steht. So ist der Rat des Verfassers nur der: den Texteditor EDLIN möglichst schnell zu vergessen und sich einen bildschirmorientierten Texteditor zu kaufen.

```
HARD DISK MANAGEMENT PROGRAM      Drive 1,  614 Cyls by  4 heads.

                    D I S K   M A N A G E R - S E A G A T E

                                Version 3.10
                            SERIAL NUMBER S00284832

        PRODUCED FOR SEAGATE BY ONTRACK COMPUTER SYSTEMS *** NOT FOR RESALE ***
                    Many prompts have HELP available by pressing F1.

                                HELP AVAILABLE BY PRESSING F1.
Only 1 drive detected, is this correct ? (y/n):

DISK MANAGER has been invoked in AUTOMATIC INSTALLATION MODE. This mode
provides the easiest way to perform most hard disk installations.  If you
wish to use options or features which are not available in AUTOMATIC MODE,
you must abort DISK MANAGER (by using the "Esc" key) and re-invoke the
DISK MANAGER program with a  "/m"  switch, as follows:

   A>dm /m      (this invokes the DISK MANAGER program in MANUAL MODE)

    DISK MANAGER(tm) Copyright(c) ONTRACK Computer Systems Inc., 1985-1987
```

Bild 7. So meldet sich der Disk-Manager

```
HARD DISK MANAGEMENT PROGRAM      Drive 1,  614 Cyls by  4 heads.

                    D I S K   M A N A G E R - S E A G A T E

                                Version 3.10
                            SERIAL NUMBER S00284832

        PRODUCED FOR SEAGATE BY ONTRACK COMPUTER SYSTEMS *** NOT FOR RESALE ***
                    Many prompts have HELP available by pressing F1.

                                HELP AVAILABLE BY PRESSING F1.
Defaulting to drive 1.
ENTER MODEL CODE FOR DRIVE 1 (RETURN if STANDARD): 4

 CODE - MODEL        CODE - MODEL        CODE - MODEL
   1  - ST213          5  - ST277R         9  - ST4053
   2  - ST225          6  - ST4026        10  - ST4096
   3  - ST238R         7  - ST4038        11  - ST4077R
   4  - ST251          8  - ST4051        12  - ST4144R
```

Bild 8. Das Festplatten-Laufwerk wird ausgewählt

Um einfache Dateien auf Diskette zu schreiben, genügt oft die Ausgabeumleitung unter MS-DOS. So können auch die Daten von der Tastatur umgeleitet werden, die nicht auf den Bildschirm, sondern in eine Datei geschrieben werden sollen. Der Befehl lautet: COPY CON Dateiname
Die nachfolgenden Tastatureingaben werden in die Datei mit dem Dateinamen geschrieben.
Bitte beachten Sie, daß Sie die Eingabe durch ein Ctrl-Z-Zeichen (Funktionstaste F6) beenden müssen. Weiter beachten Sie, daß dieser Befehl nicht prüft, ob eine Datei schon vorhanden ist und jede Neuzuordnung den alten Dateiinhalt überschreibt.

MS-DOS und Festplatten

Bisher haben wir unter MS-DOS nur mit einem oder zwei Diskettenlaufwerken gearbeitet. Nun wollen wir – sofern vorhanden – eine Festplatte formatieren und das Betriebssystem MS-DOS dort installieren. Ziel der Übung soll sein, daß in Zukunft MS-DOS direkt von der Festplatte bootet.
Üblich sind derzeit beim mc-modular-AT 20-MByte- oder 40-MByte-Festplatten (SEAGATE).
Zum Vorformatieren der 20-MB-Festplatte benötigt man das SETUP-Programm sowie den FORMAT-Befehl. Zum Vorformatieren der 40-MByte-Festplatte wird die Diskette mit dem „DISK MANAGER" benötigt.

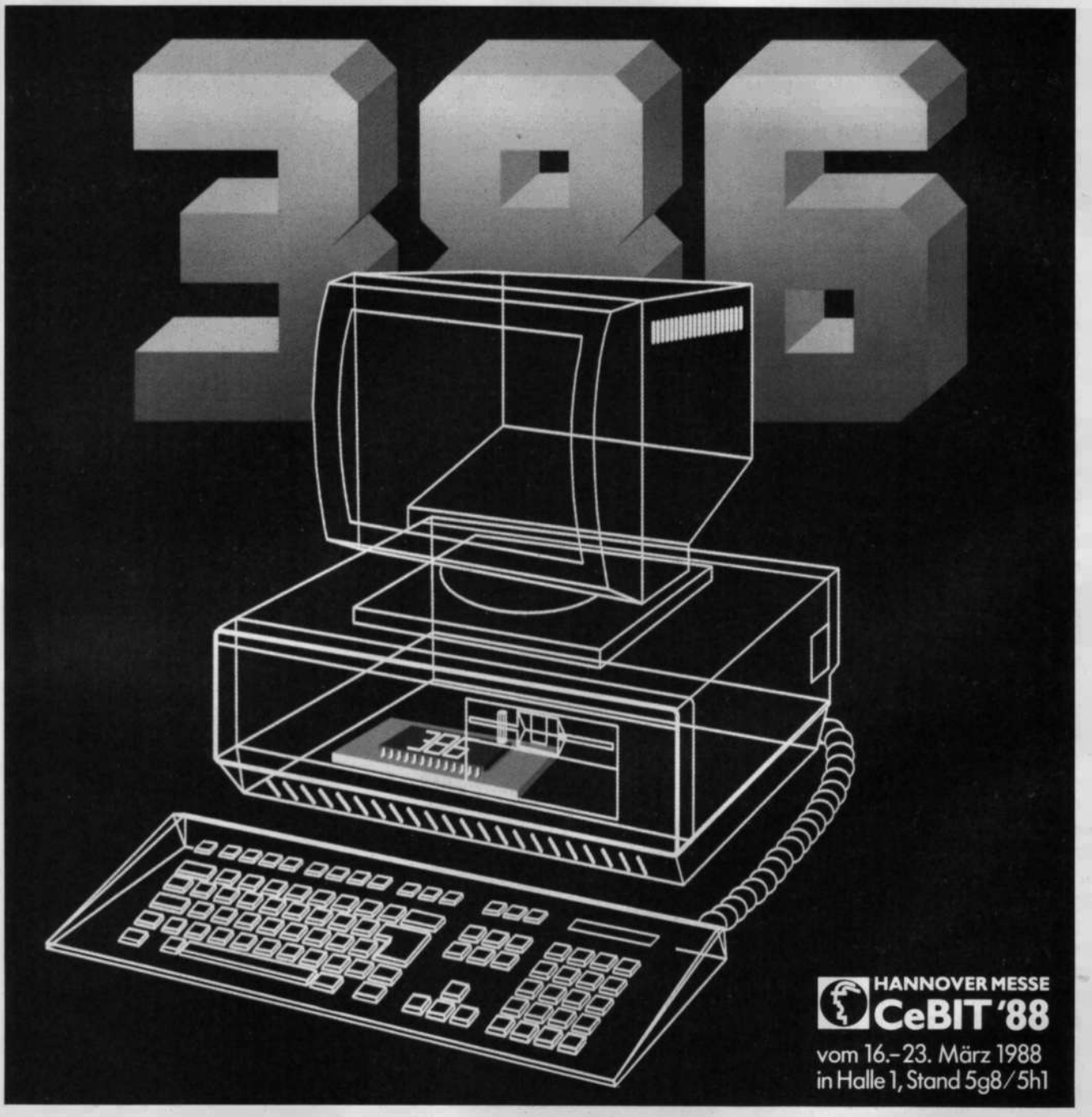
Das Vorformatieren der 20-MByte-Festplatte

Die 20-MByte-Festplatte für den mc-modular-AT heißt Seagate ST225. Das Vorformatieren ist im Handbuch des mc-modular-AT [4] detailliert beschrieben.
Fast jede Festplatte wird mit defekten Sektoren ausgeliefert, dennoch hat sie die vom Hersteller angegebene Kapazität, da bei der Fertigung der Festplatte solche Fehler bereits berücksichtigt und entsprechend mehr Sektoren aufgebracht werden.
Die fehlerhaften Sektoren werden vom Hersteller auf einem Aufkleber auf der Festplatte festgehalten und müssen beim Vorformatieren eingegeben werden. MS-DOS verwendet diese fehlerhaften Sektoren dann nicht.
Sowohl beim Vorformatieren als auch beim FORMAT-Befehl werden sämtliche Dateien der Festplatte zerstört.

Das Vorformatieren und Formatieren der 40-MByte-Festplatte

Zunächst muß man MS-DOS vom Diskettenlaufwerk aus starten; danach wird die

KONSEQUENT



Mit dem PC 60/40 bzw. PC 60/80 setzt Commodore einen Super-Mikro an die Spitze der meistverkauften PC's in Deutschland. Alle mit Industriestandard arbeitenden Mikros sind zu diesem 32 Bit Computer (16 MHz getaktet) kompatibel. Er ist gerüstet für Betriebssysteme wie OS/2 oder Unix und garantiert mit Windows/386 volles Multitasking.

Traditionell offeriert Commodore die neue PC 60 Linie mit hohem Ausstattungskomfort zu einem extrem günstigen Preis.

Ihr autorisierter Commodore Fachhändler informiert und berät Sie gerne.

Commodore –
Marktführer bei Mikrocomputern.

Commodore

Disk-Manager-Diskette eingelegt und mit DM das entsprechende Programm aufgerufen. Der Disk-Manager meldet sich (*Bild 7*) und teilt uns mit, daß er ein angeschlossenes Festplattenlaufwerk entdeckt hat und fragt uns noch einmal, ob das richtig ist. Nach der Bestätigung erscheint das in *Bild 9* abgebildete Menü. Hier muß man das entsprechende Seagate-Festplattenmodell auswählen.
In unserem Beispiel haben wir die Nummer vier – die ST251 – ausgewählt.
Die Code-Nummer 4 ist nicht unbedingt verbindlich, da es unterschiedliche Versionen des DISK-MANAGER-Programmes gibt. Wichtig ist, daß Sie die Bezeichnung der Festplatte (ST251) auswählen.
Der Disk-Manager ist recht intelligent, er kann z. B. auch fehlerhafte Konfigurationen im CMOS-RAM überschreiben, die z. B. durch ein fehlerhaftes SETUP-Programm entstanden sind.
Im Beispiel (Bild 9) hat der DISK-MANAGER den Typ 40 in das CMOS-RAM eingetragen. Danach muß noch einmal gebootet und die beschriebene Prozedur noch einmal durchgeführt werden. Anschließend werden wie beim Formatieren der 20-MByte-Festplatte die defekten Sektoren gemäß dem Aufkleber auf der Festplatte eingegeben (*Bild 10*).
Nachdem dies nun erledigt ist, steht man vor einem kleinen Problem: das Betriebssystem MS-DOS 3.2 kann Festplatten mit maximal 33,5 MByte verwalten.
Damit die restliche Kapazität der Festplatte nicht ungenutzt bleibt, kann der Disk-Manager die Festplatte in zwei logische Teile (Partitions) einteilen.
Die in *Bild 11* gezeigten Optionen A, B und C erlauben eine solche Aufteilung; die Option A ist unter MS-DOS nicht sinnvoll (aus dem oben erwähnten Grund); sinnvoll ist die Option B oder C.
Den beiden Partitionen werden aufeinanderfolgende Laufwerksbezeichnungen, bei der ersten Festplatte die Laufwerksbezeichnung C: und D: zugeordnet. Welche der beiden Optionen gewählt wird, sei dem persönlichen Geschmack und der entsprechenden Anwendung überlassen.

Der Disk-Manager zeigt zum Schluß noch einmal die gewählten Parameter (*Bild 12*). Hier besteht die letzte Möglichkeit, das Formatieren zu beenden, denn sobald man J eingibt, beginnt das System mit dem Formatieren.

Der Formatiervorgang dauert etwa fünf Minuten. Gegen Ende des Formatiervorgangs benötigt der Festplatten-Manager wieder das DOS-Betriebssystem, um automatisch

```
HARD DISK MANAGEMENT PROGRAM      Drive 1,  614 Cyls by  4 heads.

                    D I S K   M A N A G E R - S E A G A T E

                                  Version 3.10
                              SERIAL NUMBER S00284832

     PRODUCED FOR SEAGATE BY ONTRACK COMPUTER SYSTEMS *** NOT FOR RESALE ***
                  Many prompts have HELP available by pressing F1.

                                HELP AVAILABLE BY PRESSING F1.

Defaulting to drive 1.
ENTER MODEL CODE FOR DRIVE 1 (RETURN if STANDARD): 4
CMOS IS NOT CONFIGURED FOR THE IDEAL DRIVE-TYPE.
Do you want to automatically re-configure? (y/n):

                          Press Esc key to ABORT DISK INSTALLATION
```

Bild 9. Der Disk-Manager korrigiert die Konfigurationsdaten im CMOS-RAM

```
HARD DISK MANAGEMENT PROGRAM      Drive 1,  819 Cyls by  6 heads.

Cyl--Hd  Cyl--Hd  Cyl--Hd  Cyl--Hd  Cyl--Hd  Cyl--Hd  Cyl--Hd  Cyl--Hd

                          CURRENT DEFECT LIST
                                HELP AVAILABLE BY PRESSING F1.

Do you wish to enter the bad-track map for this drive? (y/n):

                          Press Esc key to ABORT DISK INSTALLATION
```

Bild 10. Defekte Sektoren muß man bei der Installierung der Festplatte angeben

das Betriebssystem sowie die Dateien COMMAND.COM sowie CONFIG.SYS auf die Festplatte zu übertragen.
Danach muß noch einmal der Festplatten-Manager aufgerufen werden, um intern benötigte Systemdateien ebenso auf die Festplatte zu übertragen.
Damit ist die 40-MByte-Festplatte formatiert und bereits bootfähig.
Beim nächsten Start des Systems – probieren Sie den Warmstart mit Crtl-Alt-Del aus – muß der Rechner nun von der Festplatte booten.
Doch halt – wir sind noch nicht ganz fertig! Alle DOS-Arbeitsdateien sind noch nicht auf der Festplatte untergebracht. Sie müssen noch von der Diskette kopiert werden. Es ist empfehlenswert, sich auf der Festplatte zunächst ein Unterverzeichnis namens „DOS" einzurichten.
Mit dem Befehl MD DOS richten Sie dieses Unterverzeichnis ein, mit CD DOS verzweigen Sie in dieses noch leere Unterverzeichnis, und mit COPY A:*.* kopieren Sie die Dateien, die auf den beiden Betriebssystemdisketten vorhanden sind.

AUTOEXEC.BAT und CONFIG.SYS

Die oben beschriebene AUTOEXEC.BAT-Datei ist nicht auf allen Betriebssystemversionen vorhanden. Falls nicht, empfiehlt es sich, diese Datei mit einem Texteditor oder mit dem Kommando COPY CON zu erzeugen.
Innerhalb der Datei CONFIG.SYS können Parameter für MS-DOS, wie die Länderspezifikation sowie die Anzahl der gleichzeitig offen zu haltenden Dateien und Puffer, spezifiziert werden. Ebenso wird hier bereits vom Disk-Manager der Typ der Festplatte eingetragen.
Anders ausgedrückt – der Disk-Manager hat bereits eine Datei CONFIG.SYS erzeugt, in welcher der Festplattentyp eingetragen ist. Die Datei CONFIG.SYS wird wie folgt ergänzt:

```
Country = 049
Files = 20
Buffers = 20
```

Die beiden letzten Eintragungen sind für dateiintensive Programme wie z. B. den Clipper-Compiler notwendig.

Bitte beachten Sie, daß MS-DOS den Inhalt der CONFIG.SYS-Datei im jeweiligen Stamminhaltsverzeichnis sucht und nur einmalig nach dem Booten liest und auswertet. Änderungen, die man hier einträgt, werden also erst nach dem nächsten Bootvorgang wirksam.

```
      HARD DISK MANAGEMENT PROGRAM      Drive 1,  819 Cyls by  6 heads.

Installing drive 1.

The formatted capacity will be  42.7 Megabytes.

It has  819 Cylinders,  6 Heads, and 17 Sectors per track.

                                   HELP AVAILABLE BY PRESSING F1.
Select one of the following options:

Option A-- 1 partition  with    .8 Megabytes, and
           1 partition  with  41.9 Megabytes
Option B-- 1 partition  with  33.5 Megabytes, and
           1 partition  with   9.2 Megabytes
Option C-- 2 partitions with  21.4 Megabytes each.

   Select an option: b

                    Press Esc key to ABORT DISK INSTALLATION
```

Bild 11. Die Festplatte wird partitioniert

```
      HARD DISK MANAGEMENT PROGRAM      Drive 1,  819 Cyls by  6 heads.

Installing drive 1.

The formatted capacity will be  42.7 Megabytes.

It has  819 Cylinders,  6 Heads, and 17 Sectors per track.

The drive will be partitioned into  1 partition  with  33.5 Megabytes, and
           1 partition  with   9.2 Megabytes.

This is your LAST CHANCE TO ABORT the installation.

****** ALL EXISTING DATA, IF ANY, ON DRIVE 1 WILL BE DESTROYED ******

Do you wish to proceed with the installation as described above? (y/n):

                    Press Esc key to ABORT DISK INSTALLATION
```

Bild 12. Nur mit der Esc-Taste kann man jetzt noch die Formatierung abbrechen

```
C:\>dir

 Datenträger in Laufwerk C hat keine Bezeichnung.
 Inhaltsverzeichnis von  C:\

COMMAND  COM    24412  15.05.87  12.00
DOS          <DIR>      3.01.80  10.25
KEYBGR   EXE     3599  15.05.87  12.00
AUTOEXEC BAT       66   3.01.80  10.30
CONFIG   SYS       35   3.01.80  10.31
        5 Datei(en)  20955136 Bytes frei

C:\>
```

Bild 13. Inhalt der Festplatte nach dem Formatieren

```
type autoexec.bat
cls
echo off
keybgr
path\;\dos;
prompt $p$g
date
time
ver
```

```
C:\>
C:\>
C:\>
C:\>
C:\>type config.sys
country=049
files=20
buffers=15

C:\>
```

Bild 14. Typische AUTOEXEC.BAT und CONFIG.SYS

MS-DOS-Funktionstasten

Funktion	Deutsche	Englische Tastatur
Befehl abbrechen .....:	STRG + UNTBR	CTRL + BREAK
Bildschirmanzeige anhalten ..:	STRG + NUM I	CTRL + NumLock
Hardcopy Bildschirm .....:	SHIFT + DRUCK	SHIFT + PrtSc
Echo auf Drucker ein/aus ....:	STRG + DRUCK	CTRL + PrtSc
Warmstart .....:	STRG + ALT + Entf	CTRL + ALT + DEL

MS-DOS-Editiertasten

Deutsche Taste	Englische Taste	Beschreibung
Einfg	Ins	Fügt Zeichen in einer Zeile ein
Entf	Del	Löscht ein Zeichen im Eingabepuffer
ESC	ESC	Löscht die gerade angezeigte Zeile; der Eingabepuffer bleibt unverändert
F1 oder ->		Zeigt bei jedem Tastendruck ein Zeichen aus dem Eingabepuffer an
F2		Zeigt alle Zeichen bis zum angegebenen Zeichen an
F3		Zeigt alle Zeichen im Puffer an; ideal geeignet, um ein Kommando zu wiederholen
F4		Löscht alle Zeichen bis zum angegebenen
F5		Nimmt die bearbeitete Zeile als aktuelle Zeile in den Puffer

Bild 15. Zusammenfassung der MS-DOS-Funktionstasten und Editiertasten

Das Formatieren der 20-MByte-Festplatte

Bei der 20-MByte-Festplatte wird kein so komfortabler Disk-Manager mitgeliefert. Es ist dies jedoch kein Hindernis – das Vorformatieren erfolgt mit der SETUP-Diskette, das Installieren der unterschiedlichen Partitionen erfolgt mit dem DOS-Befehl FDISK. Bei der 20-MByte-Festplatte muß nur eine Partition erzeugt werden.

Zum Unterschied zur 40-MByte-Festplatte muß anschließend nach dem Vorformatieren die Platte noch einmal unter MS-DOS formatiert werden; dazu verwendet man den Befehl FORMAT C:/S/V wobei die Parameter /S und /V dazu dienen, das Betriebssystem auf der Festplatte zu installieren sowie einen Versionsnamen der Festplatte zu erfragen.

Danach ist die Festplatte formatiert – den Befehl FORMAT C: sollten Sie so schnell wie möglich vergessen, um nicht versehentlich die Festplatte später noch einmal zu formatieren.

Bild 13 gibt einen Eindruck vom Directory einer so angelegten Festplatte. Die von MS-DOS benötigten Programme stehen im Stammverzeichnis, im Unterverzeichnis DOS stehen die entprechenden DOS-Programme.

Bitte achten Sie darauf, daß Sie innerhalb der AUTOEXEC.BAT-Datei den oben erläuterten PATH-Befehl eingeben, damit die DOS-Programme auch im Directory DOS gefunden werden.

Bild 14 zeigt eine typische AUTOEXEC-.BAT-Datei sowie eine CONFIG.SYS-Datei – wobei man die Befehle DATE und TIME beim mc-modular-AT vergessen kann, da dieser über eine gepufferte Uhr verfügt. Aus *Bild 15* geht die Bedeutung der MS-DOS-Funktionstasten und -Editiertasten hervor.

Nach dem Formatieren der Festplatte und dem Überspielen des Betriebssystems MS-DOS steht der uneingeschränkten Nutzung des mc-modular-ATs nun nichts mehr im Wege. Die Serie über den mc-modular-AT ist damit beendet.

Literatur:

[1] Duncan, Ray: MS-DOS für Fortgeschrittene. Vieweg-Verlag, Wiesbaden

[2] Smode, Dieter: Das große MS-DOS-Profi-Arbeitsbuch. Franzis-Verlag, München

[3] IBM Dokumentation Reihe Betriebssystem DOS

[4] Handbuch zum mc-modular-AT. Graf Elektronik, Kempten

[5] Microsoft. Handbuch MS-DOS V3.2

Michael Koch

mc-EPROM-Brenner für Z80-Systeme

Das EPROM-Programmiergerät aus Heft 1/87 hat unserem Autor so gefallen, daß er es an seinen Alphatronic-PC8 angepaßt hat. Davon können jetzt alle Besitzer von Z80-Systemen profitieren. Da im Inline-Code die Z80-Befehle im Klartext aufgeführt sind, ist die Anpassung an andere Systeme möglich.

Der Alphatronic-PC8 besitzt nur eine parallele Druckerschnittstelle. Deshalb wird das Programmiergerät mit einem zusätzlichen Port-Baustein des Typs 8255 angesteuert. Im Programm müssen nur die I/O-Routinen geändert werden. Ansonsten kann man es praktisch unverändert übernehmen.

Der 8255 wird an den Systembus des Rechners angeschlossen und über die I/O-Adressen 88H...8BH gesteuert. Port A dient zur Ausgabe der Daten an das Programmiergerät, Port B zur Eingabe der Daten in den Rechner:

88H Port A
89H Port B
8AH Port C (unbenutzt)
8BH Initialisierung

Der 8255 wird dadurch initialisiert, daß am Anfang des Programms über die Adresse 8BH die richtige Betriebsart eingestellt wird. Das geschieht am Anfang des Hauptprogramms mit der Turbo-Pascal-Zeile:

PORT[$8B] := 128 + 2;

Die übrigen Änderungen beziehen sich auf einige Konstanten (Port-Adressen), auf die Prozedur PUT_DATA und die Funktion GET_DATA, mit denen man die Daten ausgibt und in den Rechner holt. Die Daten, die in der Variablen OUT_BYTE stehen, werden bitweise seriell zum EPROM-Brenner übertragen. Genauso verfährt die Funktion GET_DATA beim Lesen des EPROMs. *Bild 1* zeigt die geänderten Routinen.

Nun muß noch die richtige Verbindung zwischen Programmiergerät und Rechner hergestellt werden (*Bild 2*). Der Anschluß ALARM wird derzeit nicht genutzt.

Rechner		EPROM-Brenner
Port	Bit	Signal
A	0	IN
A	1	CLOCK
A	2	STROBE
B	5	OUT
B	?	ALARM

Bild 2. So sind Rechner und Programmiergerät zu verbinden

Literatur

[1] *Seng, Peter:* mc-EPROM-Brenner. mc 1987, Heft 1, Seite 46...58.

```
const
        PORT_OUT = $88;
        PORT_IN  = $89;
        PORT_STR = $04;
        PORT_CLK = $02;
        NOT_PORT_CLK = $FD;
        PORT_IN_DAT  = $20;
        PORT_OUT_DAT = $01;
        PORT_DAT : byte = $00;
type
        DATA_BYTES = array [1..4] of byte;
var
        OUT_BYTE : DATA_BYTES;

procedure PUT_DATA;
begin
  inline ($06/$04/          {     LD   B ,04H          }
          $21/OUT_BYTE/     {     LD   HL,OUT_BYTE     }
          $56/              {  L1:LD   D ,(HL)         }
          $0E/$08/          {     LD   C ,08H          }
          $CB/$3A/          {  L2:SRL  D               }
          $3E/PORT_CLK/     {     LD   A ,PORT_CLK     }
          $30/$02/          {     JR   NC,S1           }
          $F6/PORT_OUT_DAT/ {     OR   PORT_OUT_DAT    }
          $D3/PORT_OUT/     {  S1:OUT  PORT_OUT,A      }
          $E6/NOT_PORT_CLK/ {     AND  NOT_PORT_CLK    }
          $D3/PORT_OUT/     {     OUT  PORT_OUT,A      }
          $0D/              {     DEC  C               }
          $20/$EF/          {     JR   NZ,L2           }
          $23/              {     INC  HL              }
          $05/              {     DEC  B               }
          $20/$E8/          {     JR   NZ,L1           }
          $3E/PORT_STR/     {     LD   A ,PORT_STR     }
          $D3/PORT_OUT/     {     OUT  PORT_OUT,A      }
          $AF/              {     XOR  A               }
          $D3/PORT_OUT);    {     OUT  PORT_OUT,A      }
end;

function GET_DATA : byte;
var
  IN_BYTE : byte;
begin
  inline ($06/$00/          {     LD   B ,00H          }
          $0E/$08/          {     LD   C ,08H          }
          $DB/PORT_IN/      {  L1:IN   A ,PORT_IN      }
          $E6/PORT_IN_DAT/  {     OR   PORT_IN_DAT     }
          $28/$01/          {     JR   Z ,S1           }
          $37/              {     SCF                  }
          $CB/$18/          {  S1:RR   B               }
          $3E/PORT_CLK/     {     LD   A ,PORT_CLK     }
          $D3/PORT_OUT/     {     OUT  PORT_OUT,A      }
          $E6/NOT_PORT_CLK/ {     OR   NOT_PORT_CLK    }
          $D3/PORT_OUT/     {     OUT  PORT_OUT,A      }
          $0D/              {     DEC  C               }
          $20/$EC/          {     JR   NZ,L1           }
          $21/IN_BYTE/      {     LD   HL,IN_BYTE      }
          $70);             {     LD   (HL),B          }
  GET_DATA:=IN_BYTE;
end;
```

Bild 1. Im wesentlichen müssen diese beiden Routinen geändert werden, damit der mc-EPROM-Brenner auch an Z80-Systemen betrieben werden kann. Die Anpassung an andere Z80-Systeme ist damit möglich

Heinrich Rescher, Eckhard Schadwinkel

Videotext in den Mikrocomputer

Videotext-Empfang mit Rechnersteuerung

Hier präsentieren wir Ihnen ein raffiniertes Interface: Das mc-Videotext-Interface. Mit diesem Zusatz können Sie die Videotexttafeln über eine serielle Schnittstelle in Ihren Rechner laden und von dort aus drucken oder weiterverarbeiten. Das Interface wird im WDR-Computerclub vorgestellt.

Ein eigener Videotext-Decoder und ein Einchip-Mikrocomputer bilden die hier beschriebene Schaltung, die über eine V.24-Schnittstelle mit jedem Computer oder Terminal verbunden werden kann. Durch diese Konfiguration ist es einerseits möglich, Videotext zu empfangen, ohne daß ein neues Fernsehgerät angeschafft werden muß. Eine solche Neuanschaffung könnte sonst notwendig werden, wenn bei älteren Fernsehgeräte keine Nachrüstdecoder lieferbar sind. Andererseits gestattet die Rechnerkopplung zusätzliche Möglichkeiten, Videotext-Seiten auszuwählen, die empfangenen Daten auszuwerten und auszugeben. Dadurch wird eine Ausgabe der Videotext-Seiten auf Drucker möglich. Denkbar ist zum Beispiel auch eine akustische Ausgabe von Texten oder Ausgabe in Blindenschrift für Sehbehinderte. Zusätzlich zur beschriebenen Schaltung wird für die Grundausbaustufe lediglich ein Fernsehempfangsteil, ein Terminal mit Anzeige und ein Fernsehgerät mit AV-Anschluß (oder ein Monitor) benötigt.

Wir haben die Erfahrung gemacht, daß der Wissensstand über Videotext sehr unterschiedlich ist. Besitzer von Fernsehgeräten ohne Videotext-Decoder haben oft nur verschwommene Vorstellungen von den Möglichkeiten, die dieses neue Medium bietet. Wir beginnen deshalb diesen Beitrag mit einer kurzen technischen Beschreibung und einem historischen Rückblick. Leser, die ausführlichere Angaben zur Technik wünschen, mögen sich gegebenenfalls an die Redaktion wenden. Bei ausreichender Nachfrage werden wir dann einen technischen Beitrag über Videotext veröffentlichen.

Was Videotext bietet

Videotext ist ein zusätzlicher Fernsehinformationsdienst, der in Deutschland gemeinsam von den Rundfunkanstalten der ARD und vom ZDF sowie als eigener Videotext-Dienst regional von den meisten ARD-Anstalten ausgestrahlt wird. Mit Videotext übermitteln die Videotext-Redaktionen Informationen an die Fernsehzuschauer. Diese Informationen können gezielt vom Zuschauer durch Eingabe einer Seitennummer abgerufen werden. Sie erscheinen dann als stehendes Bild auf einem für Videotext eingerichteten Fernsehbildschirm. Das Bild kann aus Buchstaben, Zahlen und grafischen Zeichen aufgebaut sein. Videotext kann allein oder mit dem gerade ausgestrahlten Fernsehbild betrachtet werden (Mixed Betrieb).

Die geschichtliche Entwicklung

Die Wiege von Videotext stand in Großbritannien. Dort wollten Ingenieure hörgeschädigten Zuschauern das Verständnis von Filmen durch eingeblendete Untertitel erleichtern. Eine Übertragung der Untertitel im Bild kam jedoch nicht in Frage, da sonst die nicht hörgeschädigten Zuschauern durch die Einblendung gestört würden. Die Untertitel sollten nur dann auf dem Bildschirm sichtbar werden, wenn der Zuschauer dies wünschte. Man ersann deshalb eine Ausstrahlungsmethode, die Teile des nicht sichtbaren Bildanteils nutzt. Bei Experimenten stellte man fest, daß das System zu viel mehr als nur zum Untertiteln taugt, der Weg zu Videotext war geebnet.

Das neue Verfahren wurde zum ersten Mal im Herbst 1972 angekündigt; die BBC bezeichnete ihr System als CEEFAX, abgeleitet von „to see facts", die IBA nannte ihr Verfahren ORACLE. Im Frühjahr 1973 begannen in England in unregelmäßigen Zeitabständen erste Testsendungen. Ein Jahr später, im Frühjahr 1974, einigten sich BBC und IBA auf gleiche Übertragungsdaten des Informationsdienstes, dessen offizielle Bezeichnung seitdem TELETEXT ist. Die einzelnen Textprogramm-Bezeichnungen CEEFAX und ORACLE wurden jedoch beibehalten.

Im Herbst des gleichen Jahres begann ein öffentliches Versuchsprogramm über die Sender von BBC und IBA.

Die Versuchssendungen wurden zwei Jahre lang ausgestrahlt. Am 9.November 1976 erhielt als erste Anstalt die BBC die Lizenz, CEEFAX auszustrahlen (damals noch beschränkt bis zum Sommer 1979).

Seit 1975 arbeitete das deutsche IRT (Institut für Rundfunktechnik) zusammen mit den beiden britischen Entwicklungsgesellschaften, um mit einem VHF-Sender das Ausbreitungsverhalten von TELETEXT im gebirgigen Bayern zu testen. Als diese Versuche erfolgreich abgeschlossen werden konnten, wurden IRT-eigene Versuche mit anderen Systemen zu Gunsten von TELETEXT eingestellt.

Die Kommission für den Ausbau des technischen Kommunikationssystems (KtK) schlug dann in ihrem 1976 erschienenen Telekommunikationsbericht für TELETEXT die deutsche Bezeichnung Videotext vor, um Verwechselungen mit dem Bürofernschreiben Teletex (ohne T am Ende!) zu vermeiden (Nach DIN 45060 wird das in Deutschland eingeführte System neuerdings als Fernsehtext bezeichnet. Da aber auch weiterhin der Ausdruck Videotext als Dienstbezeichnung verwendet wird, sprechen wir auch in diesem Beitrag nur von Videotext.). Auf der Internationalen Funkausstellung (IFA) 1977 präsentierten erstmals ARD und ZDF gemeinsam den neuen Informationsdienst Videotext.

Seit der IFA79 wird Videotext in Deutschland bundesweit ausgestrahlt. Es begann mit ca. 75 Informationsseiten (es wird statt Seite auch der Begriff Tafel verwendet). Man kalkulierte zu diesem Zeitpunkt die Kosten für einen Videotext-Decoder mit 2000 bis 3000 DM ein, hoffte jedoch, daß sich der Preis bei Massenfertigung erheblich senken werde. Mit Recht, denn heute schlagen die Decoderplatinen beim Kauf eines neuen Farbfernsehgeräts nur noch mit 100 bis 200 DM zu Buche.

Am 4. Oktober 1984 wurden die Parameter des Videotext-Systems als verbindlicher Standard für die Bundesrepublik Deutschland festgelegt.

Wie Videotext funktioniert

Für Videotext benötigt man keine zusätzlichen Fernsehkanäle, keine neuen Frequenzen und keine aufwendigen Sendeanlagen, denn Videotext wird als Zusatzinformation mit dem üblichen Fernsehbild ausgestrahlt. Man nutzt dafür Bildzeilen, die für das sichtbare Bild nicht benötigt werden.
Unser Fernsehsystem gestattet die Ausstrahlung von 625 Bildzeilen, von denen jedoch nur 587 das sichtbare Bild bilden. Die restlichen 38 Zeilen stellen die sog. vertikalen Austastlücken der beiden Halbbilder dar. Dadurch wird die Fernsehgeräteelektronik synchronisiert und gewinnt Zeit, den Bildwechsel vorzunehmen. In einigen dieser Zeilen wird das Videotext-Signal digital übertragen. Man nutzte anfangs nur die Zeilen 20 und 21 (Austastlükke des ersten Halbbilds) sowie die Zeilen 333 und 334 (Austastlücke des zweiten Halbbilds) für Videotext (siehe *Bild 1*). Umfangreiche Versuche hatten schon 1982 gezeigt, daß auch die Zeilen 13, 14, 326 und 327 zusätzlich für Videotext verwendet werden können, ohne Empfangsstörungen befürchten zu müssen. In den vergangenen Jahren wurde die Nutzung noch weiterer Zeilen untersucht. Denn je umfangreicher das Videotext-Angebot wird, um so wichtiger ist es, die Zusatzinformationen über möglichst viele Zeilen auszustrahlen, damit die Zugriffszeit zur einzelnen Seite akzeptabel bleibt. Zur Zeit sind für die Videotext-Übertragung bei ARD und ZDF die Zeilen 11, 12, 13, 14, 20, und 21 im ersten sowie die Zeilen 324, 325, 326, 327, 333 und 334 im zweiten Halbbild festgelegt. Das sind immerhin nun pro Vollbild 12 Zeilen, die für Videotext genutzt werden. Da je Bildzeile eine Textzeile übertragen werden kann, ergibt das eine Übertragungsrate von 300 Textzeilen je Sekunde oder ca. 12 Videotext-Seiten je Sekunde.

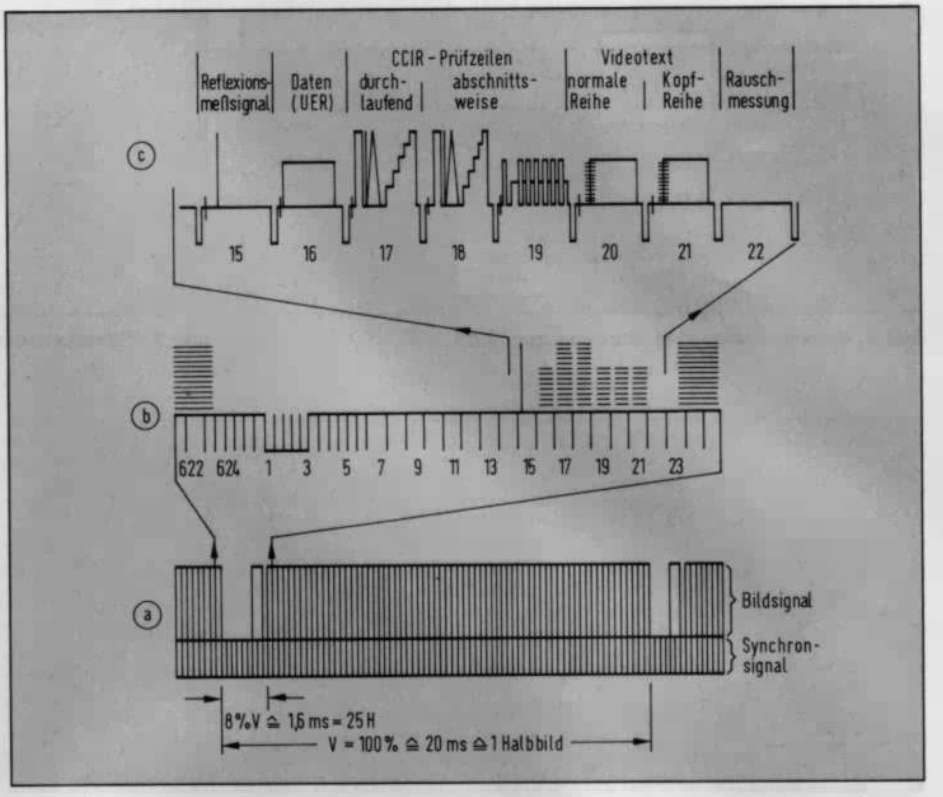


Bild 1. Übertragung von Videotext in bestimmten Bildzeilen

Beim SFB in Berlin erarbeitet täglich ein Redaktionsteam in der gemeinsamen Videotext-Redaktion von ARD und ZDF neue und überarbeitet alte VT-Seiten. Ähnlich verfährt man auch in den regionalen Videotext-Redaktionen. Da jederzeit der Seiteninhalt aktualisiert werden kann, ist Videotext wesentlich aktueller als die Printmedien. Wir empfehlen all denen, die Videotext nicht kennen, sich einmal die Sendung „Videotext FÜR ALLE" im Vormittags- beziehungsweise frühen Nachmittagsprogramm anzusehen. Diese Sendung demonstriert die Möglichkeiten von Videotext. Videotext FÜR ALLE kann mit jedem Fernsehgerät empfangen werden. Ein Videotext-Decoder ist nicht notwendig.

Jede Videotext-Seite besteht aus maximal 24 Textzeilen mit jeweils höchstens 40 Zeichen. Das ist nicht sehr viel. Der beschränkte Platz zwingt zur Kürze bei der Formulierung, zur Disziplin im Umgang mit der Sprache. Gefordert sind: Prägnanz im Ausdruck, Darstellung des Wesentlichen.
An einem Terminal werden die Seiten erstellt, die Grafiken eingegeben und die Farben bestimmt. Zumindest beim WDR sind jeder Seite sogenannte Schlüsseldaten zugeordnet, die unter anderen bestimmen, von wann bis wann die Seite gesendet werden soll. Sie bestimmen auch, wie oft die Seite in einem Sendezyklus ausgestrahlt werden soll. Alle Daten werden in einem VT-Rechner abgespeichert. Dieser steuert entsprechend der Schlüsseldaten einen sogenannten Senderechner, der wiederum die VT-Informationen im einfachsten Fall über einen Prüfzeileneinmischer dem FBAS-Signal zutastet (siehe *Bild 2*).
Die Ausstrahlung der VT-Seiten findet zyklisch statt. Je nach Wichtigkeit (wurde bei den Schlüsseldaten festgelegt), werden einzelne Seiten je Sendezyklus unterschiedlich oft ausgestrahlt (siehe *Bild 3*). Damit verkürzt sich für diese Seiten die Zugriffszeit.

Empfangsseitiger Aufwand

Videotext-taugliche Fernsehgeräte unterscheiden sich von den üblichen Geräten nur in zwei Punkten. Im Gerät selbst ist eine kleine Platine montiert, die mit einigen hochintegrierten Schaltungen bestückt ist. Dieses Modul ist der Videotext-Decoder (siehe *Bild 4*). Der zweite Unterschied findet sich bei der Gerätefernbedienung. Diese ist um einige Tasten erweitert. Sie ermöglichen es, gezielt VT-Seiten aufzurufen und besondere Videotext-Funktionen anzuwählen (Untertitel, Letzte Meldung u.s.w.). Bei den meisten Fernsehgeräten ließe sich zwar der Decoder nachrüsten, da aber die Fernbedienung oft nicht erweitert werden kann, ist der Empfang von Videotext nicht mit allen Fernsehgeräten möglich.
Im Grunde ist der Empfang von Videotext problemlos. Nur bei Reflexionen, die sich

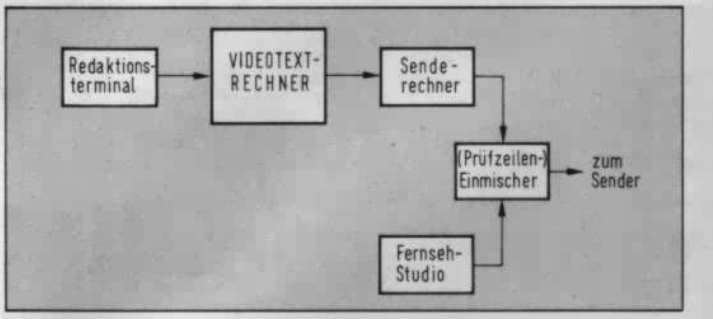


Bild 2. Videotext wird dem Bildsignal zugeführt

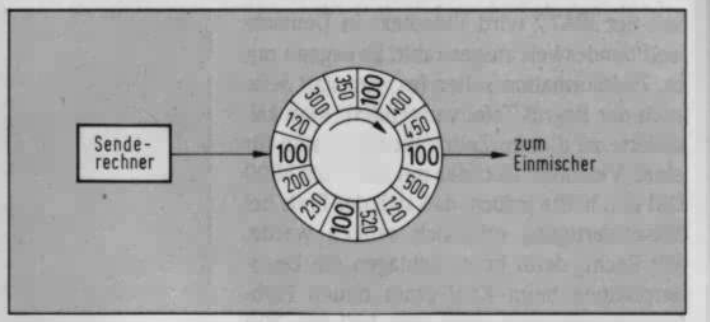


Bild 3. Videotext-Seiten werden zyklisch ausgestrahlt

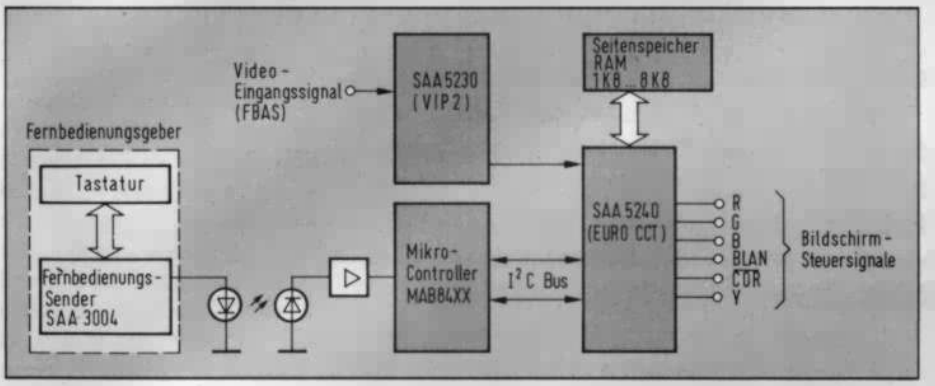


Bild 4. Prinzipschaltbild des Videotext-Decoders

im Fernsehbild als Schatten kurz hinter dem Originalbild bemerkbar machen, kann der Empfang beeinträchtigt werden. Dann müssen Änderungen an der Antennenanlage vorgenommen werden.

Was Videotext heute bietet.

Begonnen hat Videotext in Deutschland 1979 mit aus heutiger Sicht dürftigen 75 Seiten. Heute hat sich die Anzahl der Tafeln auf mehr als 300 erhöht, so daß seit der IFA87 eine neue Programmstruktur des ARD/ZDF-Videotextes eingeführt werden mußte. Die neue Gliederung gestattet nun eine noch übersichtlichere Führung durch die große Menge der vorhandenen Seiten. Die folgende Tabelle zeigt die gesamte Struktur. Für einige Kapitel (oft auch als Magazin bezeichnet) sind beispielhaft für den Umfang einige Themen angegeben (Tabelle 1). Diese Programmstruktur ist gültig für den Videotext, der über die Sender des ersten und zweiten Fernsehprogramms ausgestrahlt wird. Daneben haben viele ARD-Anstalten eine eigene Videotext-Redaktion eingerichtet, deren Seiten über die jeweiligen dritten Fernsehsenderketten ausgestrahlt werden. Die meisten Zuschauer können demnach zwischen mindestens zwei Videotext-Programmen mit unterschiedlichen Inhalten wählen.
Und noch etwas spricht für Videotext:
Videotext ist kostenlos. Es entstehen für Videotext keine zusätzlichen laufenden Kosten.

Was unsere Schaltung leistet

Zum einen kann mit ihr unabhängig vom vorhandenen Fernsehgerät Videotext empfangen werden – wichtig für Besitzer älterer Geräte.
Zum anderen gibt es völlig neue Möglichkeiten:

1. Videotext-Seiten können im angeschlossenen Computer abgespeichert, dort weiter verarbeitet und dann wieder dargestellt werden.

2. Seiten können ausgedruckt werden, wichtig unter anderem für Programmvorschauen und aktuelle Börsenberichte.

3. Da die Schaltung die VT-Daten in ein BAS-Signal umwandelt, können bei sinnvoller Beschaltung Videotext-Tafen auf Videorecorder abgespeichert werden. Ein direktes Abspeichern der VT-Daten in den Bildzeilen ist mit den üblichen Recordern nicht möglich, da die Zeilen, in denen Videotext gesendet wird, konstruktionsbedingt nicht einwandfrei aufgezeichnet werden. Mit dieser Schaltung sind nun auch hörgeschädigte Zuschauer in der Lage, die ausgestrahlten Videotext-Untertitel mit dem Bildsignal aufzuzeichnen.

4. Für Sehbehinderte ist eine Sprachausgabe oder Ausgabe in Blindenschrift denkbar.

5. Durch Erweiterung der Software wird es auch möglich sein, Innovationen des Videotext wie zum Beispiel das auf der IFA87 erstmals vorgestellte TOP-Verfahren zu nutzen.

6. Telesoftware kann wie bei Bildschirmtext empfangen werden.

So wird die Schaltung bedient

Um Ihnen zu zeigen, wie problemlos der Umgang mit der Schaltung ist, möchten wir Ihnen zunächst die Bedienung beschreiben.
Die Steuerung erfolgt über das angeschlossene Terminal oder über den angeschlossenen Computer. Wenige Befehle reichen dafür aus. Alle Befehlssequenzen müssen mit RETURN abgeschlossen werden:

Befehl <C>lear:

lösche den in der Anweisung deklarierten Textspeicher

Beispiel:
C3
F => Clear
3 => Textspeicher 3
(gültige Eingabe 0 bis 7)

Bedeutung:
Der Textspeicher 3 (befindet sich auf der beschriebenen Schaltung) soll gelöscht werden.

Befehl <F>ill:

fülle internen Textspeicher mit dem Inhalt der deklarierten Videotext-Seite. Dieser Vorgang wird anschließend ständig durchgeführt, die Seite also ständig aktualisiert!

Tabelle 1. Struktur des Videotext-Tafel-Angebotes

Tafel-Nummer	Kapitel, Themen
100	Inhalsübersicht
101-106	*Index (Stichwortverzeichnis)*
107	*VT-Redaktion*
109	*Tips für Nutzer*
110	Übersicht Aktuelles
130	VT-Interview/Zusammenfassung
140	Übersicht Wirtschaft
150	VT-Untertitel
160	Übersicht Wetter
170	Übersicht Kultur/Kritik
185-186	Medien
190	Übersicht Umwelt
191-194	*Meldungen*
195 ff	*Testtafeln*
200	Übersicht Sport
222	Letzte Meldung Sport
230-269	Tafeln für EM, WM, Olympiade etc.
270	Fußball aktuell
280-281	Basketball
282-283	Eishockey
284-285	Handball
286	Schach
287-288	Tischtennis
289-290	Volleyball
291	Wasserball
300	Übersicht Fernsehen
301	Fernsehen heute
321	Fernsehen morgen
341	Fernsehen übermorgen
360	Übersicht für eine Woche
390	Übersicht Programmbegleitung
399	Programmänderungen ARD/ZDF
400	Übersicht Andere Programme (Hörfunk + Fernsehen)
410	Übersicht III. Fernsehprogramme
450	Übersicht Hörfunk
460	GEZ
500	Übersicht Service
510	Übersicht VT-Ratgeber
520	Warentest
521	*Heizölpreise*
530	Übersicht Kulturtips
540	Übersicht Gewinnzahlen
550	Übersicht Verkehr/Reise
560	Übersicht Gesundheit
561	*AIDS*
565	*Pollenflug (April-Juli)*
569	*Gesundheitstips*
570	Übersicht Spaß und Spiel
580	Lesen statt Hören
599	Weltzeituhr
600	Übersicht Presse

Beispiel:
F0670
F => Fill
0 => Textspeicher 0
(gültige Eingabe 0 bis 7)
670 => Videotext-Seite 670

Bedeutung:
Der Textspeicher 0 soll ständig mit den Daten der Videotext-Seite 670 gefüllt beziehungsweise aktualisiert werden.

Hinweis:
Der EURO CCT kann konstruktionsbedingt nur vier Seiten zur gleichen Zeit aktualisieren. Auch ist es nicht erlaubt, die gleiche VT-Seite auf mehr als einen Textspeicher abzuspeichern.

Befehl <D>isplay:

Der aktuelle Inhalt des deklarierten Textspeichers soll auf dem angeschlossenen Fernsehgerät erscheinen.

Beispiel:
D2
D => Display
2 => Textspeicher 2

Bedeutung:
Auf dem angeschlossenen Fernsehgerät wird der Inhalt des Textspeichers 2 abgebildet.

Befehl <A>rrived:

Kontrolliere, ob ein deklarierter Textspeicher mit Videotext-Daten gefüllt ist beziehungsweise, ob die über F gesuchte Seite gefunden und abgespeichert wurde. Befinden sich im Textspeicher Daten, erscheint am Rechnerdisplay das Zeichen *, ist der Speicher leer, erfolgt keine Reaktion. Die Kontrolle wird laufend fortgesetzt. Die Funktion kann durch eine beliebige Tastenbetätigung abgebrochen werden.

Beispiel:
A7
A => Arrived
7 => Textspeicher 7

Bedeutung:
Der Textspeicher 7 wird überprüft.

Die folgenden beiden Befehle wirken sich nur aus, wenn der Fernsehbildschirm sein Eingangssignal als FBAS-Signal erhält. Wir haben in den FBAS-Weg einen elektronischen Videoschalter eingebaut, den die Befehle PICTURE und TEXT ansteuern.

Befehl <P>ICTURE:

Das übliche Fernsehbild soll dargestellt werden.

Beispiel:
P
P => Picture

Tabelle 2. Leistungsmerkmale des Zeichengenerators

- Zeichenwiedergabe mit 12*10-Rasterpunkte
- Betrieb mit oder ohne Zeilensprung über die Software wählbar
- ausschließlich serielle Attribute
- 128 alphanumerische und grafische Zeichen
- 3 Sprach-Zeichensätze (Englisch, Deutsch, Schwedisch)
- 6 Sonderzeichen (Symbole)
- Open-Drain-Ausgänge für R, G, B und Y
- Open-Drain-Ausgänge für Austastung (BLAN) und Kontrastreduzierung (CORnicht)
- Y-Ausgang nur in der Zeichenebene wirksam
- Y-Ausgang ohne Blinkfunktion z.B. zur Druckeransteuerung
- BLAN-Ausgang erlaubt Zeichen-, Umfeld- und Vollbildtastung
- CORnicht-Ausgang erlaubt softwaregesteuerte Kontrastreduzierung für Mischbetrieb (superimpose = direkte Zeicheneinblendung) und Umfeldaustastung

Tabelle 3. Leistungsmerkmale des EURO CCT

- Standard-I²C-Bus-Sender/Empfänger
- Arbeitsfrequenzbereich 0 bis 10 kHz
- Übertragungen werden quittiert
- Autoinkrementierung der Positionsregister nach bestimmten Steuerbefehlen, direkte Adressierung ebenfalls möglich
- Autoinkrementierung bei bestimmten Steuerregistern, auch hier direkte Adressierung möglich
- alle RAM-Speicherplätze über den I2C-Bus für Lese- und Schreiboperationen zugänglich

Bedeutung:
Der Bildschirm des Fernsehgeräts zeigt das Fernsehbild.

Befehl <T>EXT:

Der Videotext soll angezeigt werden.

Beispiel:
T
T = > Text

Bedeutung:
Am Bildschirm des Fernsehgerätes wird das VT-Bild sichtbar. Die folgenden drei Befehle verhalten sich wie Schalter, sie schalten zwischen zwei Zuständen hin und her. Der gekennzeichnete Buchstabe muß mit der Control-Taste zusammen betätigt werden.

Befehl: TOGGLE<G>RAFIK

Grafiksymbole beim Auslesen unterdrükken, ja/nein.

Beispiel:
Ctrl. G
In Abhängigkeit vom vorhergehenden Zustand Grafiksymbole beim Auslesen unterdrücken oder Grafiksymbole beim Auslesen nicht unterdrücken.

Bedeutung:
Je nach „Schalterstellung" erscheinen am Computer-Bildschirm die gesendeten Grafiksymbole oder werden als Leerzeichen dargestellt. Da der Computer üblicherweise nicht in der Lage ist, die Steuersignale für Videotext-Grafikzeichen auszuwerten, kann es sinnvoll sein, diese Zeichen am Bildschirm nicht anzuzeigen (Grafikzeichen sind bei Videotext übliche Textzeichen, die durch ein vorangestelltes Sonderzeichen als Grafikzeichen gekennzeichnet werden. Alle folgenden Zeichen werden solange als Grafikzeichen interpretiert, bis ein weiteres Sonderzeichen diesen Mode ausschaltet oder das Zeilenende erreicht wird.). Anstelle der Grafikzeichen werden (Leerstellen dargestellt. Damit wird erreicht, daß der Zeilenumbruch beibehalten wird. Da Videotext kein Zeilenende-Signal kennt, würde sich sonst der Seitenaufbau erheblich verändern (default: Grafikzeichen anzeigen).

Befehl TOGGLE <X>ON/XOFF:

Verändern des Übertragungsprotokolls

Beispiel:
Ctrl. X
Toggle XON/XOFF

Bedeutung:
Das Übertragungsprotokoll der seriellen Schnittstelle wird auf XON/XOFF geschaltet beziehungsweise arbeitet ohne Protokoll (default: XON/XOFF).

Befehl TOGGLE S<T>ATUSZEILE:

Statuszeile anzeigen ja/nein

Beispiel:
Ctrl. T
Toggle Statuszeile

Bedeutung:
Der EURO CCT bietet die Möglichkeit, am oberen oder unteren Bildrand eine zusätzliche Statuszeile einzublenden. Der Mikroprozessor nutzt diese Möglichkeit und gibt in dieser Zeile den Status über Grafik-Mode und XON/XOFF-Betriebsart an.
Die Statuszeile, die sich üblicherweise am oberen Bildrand befindet, wird durch Ctrl. T ein- oder ausgeschaltet (default: ohne Statuszeile).
Die folgenden beiden Befehle dienen zur Datenübertragung zwischen Videotext-Decoder und angeschlossenen Rechner.

Befehl <R>ead:

lese die Daten aus dem deklarierten Textspeicher aus und übertrage sie an den angeschlossenen Rechner. Durch die Eingabe eines zweiten Buchstabens kann bestimmt werden, ob das kontinuierlich erfolgen soll:
<C>ontinue:
wiederhole den gesamten Vorgang (Auslesen und Übertragen) immer dann, wenn die Seite neu (innerhalb des aktuellen Sendezyklus) übertragen wird.

<S>ingle:
führe den gesamten Vorgang nur einmal durch.
Nach Deklaration des Textspeichers kann bestimmt werden, ob jeweils die gesamte Seite oder nur ein festzulegender Ausschnitt übertragen werden soll:
<>:
Die ganze Seite wird übertragen
<adresse>:
adresse ist eine achtstellige Zahl, die den Seitenausschnitt bestimmt. Die Angabe muß in dieser Form erfolgen: erste auszulesende Zeile, Anzahl der auszulesenden Zeilen, erste auszulesende Spalte, Anzahl der auszulesenden Spalten (die Angaben haben jeweils zweistellig, direkt aneinander gereiht zu erfolgen).

Beispiel:
RC5
R => Read
C => continue
5 => Textspeicher 5

Bedeutung:
Übertrage den gesamten Inhalt vom Textspeicher 5 kontinuierlich zum Rechner.

Beispiel:
RS302150832:
R => Read
S => single
3 => Textspeicher 3
02 => erste auszulesende Zeile 02

15 => 15 Zeilen sollen ausgelesen werden
08 => ab Spalte 08 soll ausgelesen werden
32 => 32 Spalten sollen übertragen werden

Bedeutung:
Die Seite, die im Textspeicher 3 vorliegt, soll ab Zeile 2 bis Zeile 16 von Spalte 8 bis Spalte 39 nur einmal übertragen werden.

Befehl <WD> write data:

Übertrage Daten vom angeschlossenen Rechner in den deklarierten Textspeicher der Schaltung. Nach Eingabe der Nummer des Speichers kann wie beim Befehl READ entschieden werden, ob die ganze oder nur ein Teil der Seite übertragen werden soll.
<>:
Die ganze Seite wird übertragen
<adresse>:
adresse ist eine achtstellige Zahl, die den Seitenausschnitt bestimmt. Die Angabe muß in dieser Form erfolgen:
erste auszulesende Zeile
Anzahl der auszulesenden Zeilen
erste auszulesende Spalte
Anzahl der auszulesenden Spalten
(die Angaben haben jeweils zweistellig, direkt aneinander gereiht zu erfolgen).

Beispiel:
WD4
WD => write data
4 => Textspeicher 4

Bedeutung:
Die Daten im Rechner werden komplett in den Textspeicher 4 übertragen.

Besonderen Anwendungsfällen dient der

Befehl <WI> (write instruction).

Dieser Befehl ermöglicht die direkte Steuerung des Euro-CCT. Damit können alle Möglichkeiten dieses Bausteins direkt genutzt werden. Die Anwendung des Befehls erfordert Kenntnisse des CCT-Bausteins. Da hier die Möglichkeit bestehen soll, alle möglichen Zeichen, also auch CR, zum CCT zu schicken, muß in einer besonderen Weise eingegeben werden. Nach Eingabe der Befehlskennung WI muß angegeben werden, wieviel Zeichen folgen werden, also noch zu diesem Befehl gehören. Dann wird zunächst mit RETURN abgeschlossen. Anschließend werden die vorgesehenen Zeichen nacheinander eingegeben, ohne mit RETURN abzuschließen.

Erläuterungen:
Der Euro-CCT besitzt elf Register. Davon können Register R1 bis R10 nur beschrieben werden. Register R11 kann sowohl geschrieben als auch gelesen werden. Um das Programmieren zu vereinfachen, ist von Valvo eine Autoinkrementierung der meisten Steuerregister implementiert worden. Das bedeutet, daß bei bestimmten Befehlssequenzen, die nacheinander mehrere Register ansprechen, nicht jedesmal die nächste Registernummer definiert werden muß; der CCT „ahnt", welches Register beim nächsten Befehl gemeint ist. Über jedes Register können bestimmte Funktionen angewählt werden:

Register 1: Setzen der Betriebsart
(AI zu Register 2)
Register 2: Seitenspeicher-Steuerung
(AI zu Register 3)
Register 3: Suchseiten-Steuerung
Register 4: Anzeige der Seitenspeicher
(AI zu Register 5)
Register 5: Anzeigensteuerung für den Normalbetrieb (AI zu Register 6)
Register 6: Anzeigensteuerung für Blitzmeldungen und Untertitel (AI zu Register 7)
Register 7: Aufbau des Anzeigefeldes
Die Register 8 bis 10 adressieren die Zeichenplätze im Textspeicher:
Register 8: Wahl des Seitenspeichers, Löschen ja/nein (AI zu Register 9)
Register 9: Reihenadresse
(AI zu Register 10)
Register 10: Spaltenadresse
(AI zu Register 11)
Register 11: Lesen/Schreiben der α/N-Codes (Code, der die Videotext-Zeichen erzeugt), aus der bzw. in die Speicherstelle, die in den Registern 8, 9 und 10 deklariert wurde.
(AI steht für Autoinkrementierung.)

Es ist zu umfangreich, alle Funktionen im Detail zu beschreiben. Viele Möglichkeiten wird man auch nicht über den Befehl WI anwählen, da die anderen, eingebauten Befehle diese Funktionen bereits nutzen. Wir möchten deshalb an zwei Beispielen erläutern, wie Funktionen genutzt werden, die aus unserer Sicht wichtig sind und nicht über die eingebauten Befehle direkt angesprochen werden können. Die gesamte Aufstellung kann dem Datenblatt zum Euro-CCT SAA 5240 A/B der Firma Valvo entnommen werden.

Beispiel 1: Zeilensprung ja/nein
Damit das übliche Fernsehbild möglichst flimmerfrei am Bildschirm betrachtet werden kann, werden die zu übertragenden Bilder in zwei ineinander geschachtelte Halbbilder aufgeteilt und nacheinander übertragen. Bei stehenden Bildern wie VIDEOTEXT ist dieser „Trick" jedoch nicht so wirkungsvoll. Der Euro-CCT besitzt daher die Möglichkeit, bei Videotext den Zeilensprung nicht vorzunehmen und damit das Bild ruhiger erscheinen zu lassen. Die Wahl dieser Betriebsarten wird mit Register 1 vorgenommen.

Beispiel:
WI04(RETURN)
WI => write instruction
04 => vier Zeichen sollen folgen. Je zwei bilden ein Byte, so daß zwei folgende Bytes zum Befehl gehören

0106
01 => erstes Byte, kennzeichnet das Register mit dessen Nummer in hexadezimaler Schreibweise.
06 => zweites Byte, Wahl der Betriebsart, 06 in hexadezimaler Schreibweise läßt sich in binärer Schreibweise als 0000 0110 darstellen. Jedes Bit dieses Ausdrucks hat laut Datenblatt eine besondere Bedeutung:
0 => kein Test
0 => Übertragung mit Paritätsbit
0 => Normalbetrieb
0 => keine virtuellen Reihen
0 => kein „full channel mod"
1 => TCS ein
1 => siehe unten
0 => siehe unten
Diese beiden letzten Bits müssen zusammen betrachtet werden:
10 => kein Zeilensprung 312/312 Zeilen

Bedeutung:
Im Normalbetrieb des Videotext soll kein Zeilensprung stattfinden. Es werden zweimal 312 Zeilen (ohne Halbzeilen) dargestellt. Diese Betriebsart wird zwangsläufig bei Blitzmeldungen und Untertitelung wegen der gleichzeitigen Darstellung von Bild- und Textzeilen unterdrückt (= nicht realisiert).

Beispiel 2: doppelte Zeichenhöhe
Der EURO CCT bietet die Möglichkeit, die Zeichen der Videotext-Seiten in doppelter Höhe am Bildschirm abzubilden. Wegen des doppelten Platzbedarfs kann dann natürlich nicht die gesamte Seite zur gleichen Zeit betrachtet werden. Es muß deshalb angegeben werden, ob die untere oder obere Seitenhälfte am Bildschirm erscheinen soll.

Beispiel:
WI04 (RETURN)
WI => write instruction
04 => vier Zeichen, entsprechend zwei folgenden Bytes, gehören zum Befehl

078F
07 => erstes Byte, Registernummer

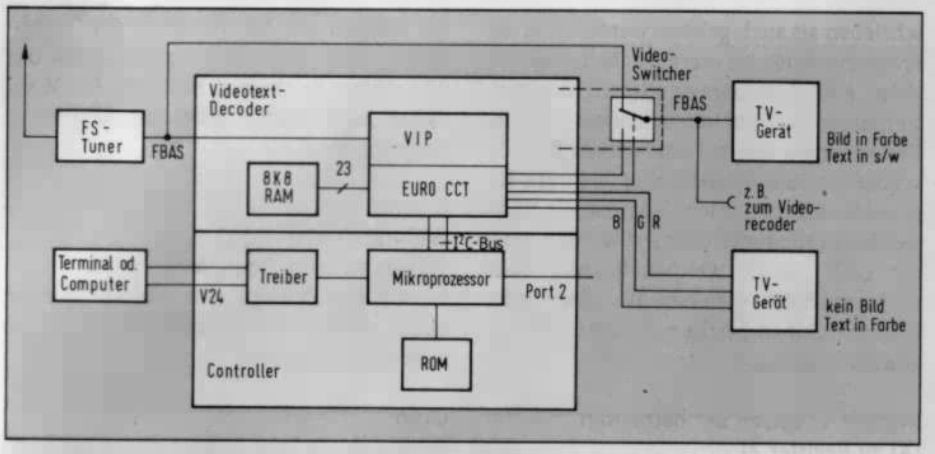


Bild 5. Blockschaltbild der gesamten Schaltung

8F => zweites Byte, Festlegung des Anzeigefeldes
8F (hex = 1000 1111 (bin)
Jedes Bit hat eine besondere Bedeutung. Wir beginnen bei der Erklärung wieder beim höchstwertigen Bit:
1 => Status-Information oben (Reihe 0)
0 => kein Cursor
0 => verdeckte Anzeige möglich
0 => obere Bildhälfte
1 => doppelte Zeichenhöhe
1 => Boxattribut in Reihe 24
1 => Boxattribut in den Reihen 1 bis 23
1 => Boxattribut in Reihe 0

Bedeutung:
Die Zeichen der oberen Hälfte der VIDEOTEXT-Seite sollen in doppelter Höhe dargestellt werden.
Damit sind alle, in die Software implementierten Befehle beschrieben. Wir möchten nun noch beispielhaft einen kompletten Bedienablauf beschreiben:

Wir verwendeten als Terminal einen Rechner Olivetti M10, betrieben im Terminal-Mode. Dessen V24-Schnittstelle hatten wir auf 9600 Baud, 8 Datenbits, no parity, 1 Stop-Bit und XON/XOFF enable eingestellt.
Zunächst werden alle Geräte eingeschaltet. Bei der Initialisierung werden alle acht Textspeicher auf der Platine gelöscht. Am Bildschirm erscheint eine Meldung, die unter anderem auffordert, eine Taste zu betätigen. In der Zwischenzeit veranlaßt das Programm, daß die VIDEOTEXT-Seite 100 angewählt und in den Speicher 0 abgelegt wird.
Wird nun eine beliebige Taste betätigt, erscheint die Seite 100 am Fernsehbildschirm. Dieser Seite kann z. B. entnommen werden, daß die Seite 103 das „Fernsehprogramm heute abend" zeigt. Die nächsten Schritte bewirken, daß diese Seite am Fernsehbildschirm erscheint und anschließend, ohne Grafikzeichen, am Display des Rechners betrachtet werden kann.

F1303
Seite 303 soll in den Speicher 1 geladen werden.
D1
Der Inhalt von Speicher 1, hier VT-Seite 303, erscheint am Fernsehbildschirm.

Man sieht, daß die Seite Grafik-Zeichen enthält. Wir entscheiden uns, bei der Übertragung zum Rechner die Grafik-Zeichen in Leerzeichen umzuwandeln. Um den Status des Grafik-Schalters direkt zu erkennen, schalten wir zunächst die Statuszeile ein.

Ctrl. T
Statuszeile ein.
Es erscheint dort unter anderem die Meldung „Grafik ON".

Ctrl. G
schaltet auf „Grafik OFF".
RS1
bewirkt, daß der Inhalt von Speicher 1, hier die VT-Seite 303, zum Rechner übertragen wird. Die Seite erscheint auf dem Rechnerdisplay ohne Grafikzeichen. Nun kann sie abgespeichert, weiter bearbeitet oder ausgedruckt werden.

Die Schaltung

Bild 5 zeigt das Blockschaltbild. Es zeigt zwei Funktionsblöcke: den eigentlichen Videotext-Decoder und den Mikrocontroller.

Der Videotext-Decoder

Ein Fernsehtuner (selbstständiges Empfangsteil oder eingebaut in ein Fernsehgerät) liefert ein Videosignal (FBAS). Dieses Signal wird der Schaltung zugeführt. Der Decoder tastet die Videotext-Daten aus dem Videosignal aus, bereitet sie auf und erzeugt daraus die Daten für die einzelnen Textseiten. Insgesamt acht Seiten (numeriert von 0 bis 7) können im Speicherbaustein (RAM) abgelegt werden. Außerdem erzeugt der Decoder verschiedene Videoausgangssignale zur Darstellung der Videotext-Seiten auf einem angeschlossenen Fernsehgerät (oder Monitor). Für den MIX-Betrieb (Fernsehbild mit überlagerten Videotext) und für den BOX-Betrieb (Textzeilen werden anstelle des Bildsignals mit schwarzem Hintergrund eingeblendet) ist ein Videoschalter vorgesehen. Er wird auch genutzt, um Untertitel in Schwarz/Weiß mit dem Farbbildsignal auf Videorecorder abzuspeichern. (Da die heutigen Videorecorder zur Aufnahme ein FBAS-Signal benötigen, muß das Videotext-Signal auch in dieser Form bereitgestellt werden. Videotext in Farbe liegt jedoch als RGB-Signal vor. Eine Umwandlung von RGB in FBAS ist zwar möglich, befriedigt in der Darstellungsqualität jedoch aus folgendem Grund nicht: EURO CCT und RGB/FBAS-Decoder laufen nicht phasenstarr; es entstehen daher Farbsäume, die mit aufgezeichnet würden.)

Der Mikrocontroller

Der Mikrocontroller (Mikroprozessor) steuert den oben beschriebenen Decoder. Beide Funktionsblöcke kommunizieren miteinander über den sogenannten I^2C-Bus. Das Protokoll dieser Busschnittstelle ist genormt. Daher kommt als Controller prinzipiell jeder Mikroprozessor in Frage. Das Hauptproblem besteht darin, die Software I^2C-Bus-gerecht zu erstellen. Die Software unseres Mikroprozessors ist in einem EPROM abgelegt.
Die Steuerung der gesamten Schaltung erfolgt über ein angeschlossenes Terminal oder über einen angeschlossenen Computer. Mikrocontroller und Terminal kommunizieren über eine V24-Schnittstelle.

Wird fortgesetzt

Literatur

Videotext im Aufwind, Funkschau 22/1985
Videotext – Oracle/Ceefax – Antiope, Alexander Kulpok, Berlin 1979
Videotext in der Bundesrepublik Deutschland, Fernseh- und Kino-Technik, 5/1984
Datenblatt SAA 5240 A/B, Firma VALVO, 9.85
Technische Information ...I^2C-Bus..., TI 811215, Firma VALVO
Technische Information ...Decoder..., TI 840314, Firma VALVO
I^2C-bus specification, Firma VALVO
I^2C-bus software examples, Firma VALVO
Computer Controlled Teletext User's Manual, Firma VALVO
Faltblatt ARD/ZDF-Videotext Programmübersicht, Berlin 1987

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

3", 3,5", 5,25", 8"
SS, DS, SD, DD, HD
PC/XT/AT
360/720 KB – 1,2/1,44 MB

Max. 4 Drives, gem. Betrieb, Umschaltung hardware- oder softwaremäßig (Option). Anschlüsse: 34pol. Standard und 50pol. 8". Kurze Karte. Lesen/Schreiben der Formate 360/720/1200/1440 KB (AT/S2-Format) mit mitgelieferter Software. Fremdformate mit Spezialsoftware.

Der Supercontroller für die Datenkonvertierung DM 495.–

MITSUBISHI-Laufwerke

MF355C, 3,5", 1/2 MB DM 280.–
M2896, 8", 1,6 MB DM 1200.–
MF504, 5,25"-Multifunktionslaufwerk, 40/80 Track, 0,5/1/1,6 MB DM 270.–

WEGE ELEKTRONIK

Hans-Detlef Wege
Grubenstraße 4, 4130 Moers 3,
Telefon bis 19 Uhr 0 28 41/7 20 38

Leistung überzeugt!
Diese Preise stimmen!

PC 80386, 6/16 MHz, 1 MB RAM, 8 Slots, Uhr und Kalender, **H.E.G.A.-Karte,** ser./par., 1,2-MB-Floppy, 20-MB-Festpl., 200-W-Netzteil, Klappgeh., deutsche Tastatur mit 102 T., leg. DOS/BIOS, nur ..... **DM 5350**
Mit 40-MB-Festpl., 28 ms ..... DM 5950

Auch 80286 enorm günstig, ab ..... **DM 1998**

XT, 640 KB, 2 Fl., 10 MHZ, 102er-T. .... **DM 1698**
mit nur 1 Floppy ..... **DM 1548**
Steckkarten, Mäuse, Monitore günstig! Z. B.
E.G.A. 800 × 600 nur ..... **DM 399**
AT 2,5-MB-card, 0 K ..... **DM 199**
30-MB-Festplattenset mit Controller und Kabelsatz ..... DM 695
14" EGAMonitor ..... DM 990

Fölsche Computer

Wolfgang Fölsche · Hard- und Software
Untertorstraße 24 · 7022 L.-Echterdingen
Telefon 07 11/79 32 93

PCs und sämtliches Zubehör

TELESERVICE

Programmpaket für Software-Fernwartung, 300/1200 Bd, für Akustikkoppler oder Modem, ideal für Software- und Systemhäuser. Ferngesteuertes Ausführen von DOS-Befehlen und Anwenderprogrammen auf dem Rechner des Kunden, Datei-Transfer und Directory-Anzeige im Hintergrund (d. h. der Kunde kann währenddessen weiterarbeiten!).

- Erhebliche Reisekosten-Ersparnis
- Ferngesteuerte Tastatureingabe-Simulation
- Übertragen von Bildschirm-Inhalten im Textmodus
- Dadurch z. B. dBASE-Fernsteuerung möglich (!)
- Speicherresident, läuft im Hintergrund (25 KByte)
- Weitergabe an alle Ihre Kunden ohne Zusatzkosten

Teleservice-Programm (für den Rechner des Kunden), passendes Terminal-Programm mit Dateitransfer (für Sie), deutsches Handbuch ..... **485 DM**

shamrock software

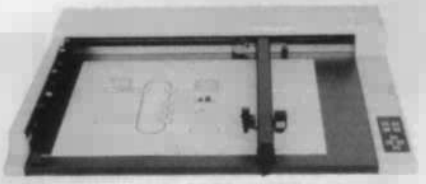
SHAMROCK SOFTWARE GMBH

Tel. 0 89/3 08 17 43, Klausingweg 6, 8000 München 40

ÖSTERREICH: PCS, Tel. 02 22/94 41 79, Linzer Str. 210, A-1140 Wien

SCHWEIZ: Microtron, Tel. 0 32/87 24 29, Bahnhofstr. 2, CH-2542 Pieterlen

A3-PLOTTER



MAXY-8 komplett mit 8 Stiften

- HP-GL-kompatibel
- 25 µm Schrittweite
- 8 Stifte
- 400 mm/s
- serielle und parallele Schnittstelle

DM 2349.–

W&O Computertechnik

Heinrichstraße 2, 4390 Gladbeck

Telefon (0 20 43) 2 40 11, Telex 8 579 202 WOCD

RAMs EPROMs EPROMs RAMs

Dyn. RAMs		
4164-12 (120 ns)	NEC	DM 4,79
4164-10 (100 ns)	FUJITSU	DM 6,95
41256-15 (150 ns)	NEC	DM 7,75
41256-12 (120 ns)	SHARP	DM 8,27
41256-10 (100 ns)	OKI	DM 10,72
41256-8 (80 ns)	NMB	DM 17,78
Stat. RAMs		
µPD 4016 C3 (150 ns)	NEC	DM 4,23
HY 6264 LP10 (100 ns)	HYUNDAI	DM 6,61
µPD43256-12L (120 ns)	NEC	DM 23,37
EPROMs		
2764-25	TEXAS INSTR.	DM 9,46
27128-25	SGS	DM 9,92
27256-20	NEC	DM 13,68
CMOS-Mikroprozessoren		
V20 (8 MHz)		DM 15,79
V30 (8 MHz)		DM 21,57
Numerische Coprozessoren		
8087 - 80287 - 80387		auf Anfr.
Festplatten-Laufwerke von Seagate		
ST 225 (21,4 MB, 65 ms)		DM 549,48
ST 238R (32,7 MB, 65 ms)		DM 562,–
ST 238R inkl. Controller		DM 718,20
ST 251 (42,8 MB, 40 ms)		DM 981,54
ST 252/1 (42,8 MB, 28 ms)		DM 1179,90
ST 125 (21,4 MB, 35 ms)		DM 681,72
ST 157R inkl. Controller (49,1 MB, 30 ms)		DM 1113,78

Angebot freibleibend!

MEMORY ELECTRONICS
Inh. Ursula Nohe, Dechsendorfer Straße 10,
8522 Herzogenaurach, Telefon 0 91 32/6 11 61

VIDEODIGITALISIERER
f. IBM PC, XT/AT u. Kompatible

VD 1951: Schneller, hochauflösender 1-Bit-Echtzeitdigitalisierer. Scanfrequenz 12 MHz. Digitalisierung in 1/50 s. Hochwertiger 70-MHz-Eingangsverstärker. 3 Eingangskennlinien, 2 Graumuster (grob u. fein) über Tastatur wählbar. Kontrast, Helligkeit, Graustufen (max. 5) stufenlos regelbar. Auflösung: Hor. 770 Pixel, Ver. 311 Zeilen. 512 KBit RAM on Board. Option: Externe Triggerung. Karte belegt 1 Slot. Umfangreiche Softwareunterstützung in Turbo-Pascal (teilw. Sources) für IBM-CGA, Hercules- und EGA-Grafikkarte.
Best.Nr: VD 1951 (Digitalisierer u. Software) DM 2100,–

VD 2000/8: 8-Bit-Echtzeitdigitalisierer. 256 Graustufen. Input Look-Up-Table (LUT), Ausgangsfarbgenerator (CLUT). 256 KByte RAM on Board. Alle anderen Daten wie VD 1951. Umfangreiche Softwareunterstützung in Turbo-Pascal (teilw. Sources) für Hercules- und EGA-Grafikkarte.
Best.Nr.: VD 2000/8 (Digitalisierer u. Software) DM 4100,–

Info gratis, Demo-Disk DM 20,– (V-Scheck od. Schein).

DIPL.-ING. OTTMAR WEGNER
Adalbertstraße 25, 2300 Kiel 1, Telefon 04 31/33 38 77

Cross-Assembler

Professionelle Cross-Assembler für PC-DOS/MS-DOS und VAX-Systeme, MACROs, Linker, Library-Manager, Conditionals, unbegrenzte Programmlänge, verschachtelte Strukturen, schnelle Compilation, zahlreiche Zielformate.

Unterstützte Prozessor-Familien:

Z8, Super-8, Z80, Z180, 64180, Z280, Z8000, Z80.000, 8080, 8085, NSC800, 8088/86, 80186/286, 8048, 8044/51, 80452, 80515, 8096, 83c351, 8400, 1802, 6301, 6800/02/08, 6801/03, 6804, 6805, 6809, 68000/8/10, 68020, 65c01/02/11, 65c816, 32000, V20/V30, F8/3870, NEC7500, NCR32

Ingenieurbüro für Datenverarbeitung
Dipl.-Ing. Hans-Jürgen Wilke
Adalbertsteinweg 26 D-5100 Aachen
Tel.: 0241/542228 FAX: 0241/533230

In-Circuit-Emulation

Neue Maßstäbe hinsichtlich Preis und Leistung setzen NORAL-In-Circuit-Emulatoren:

- Multi-Prozessor-fähig
- Komplexe, verschachtelte Break- und Trigger-Bedingungen
- Leistungsstarke Trace-Funktion
- Symbolisches Debuggen
- Overlay-Speicher bis 1 MByte
- Terminal-, PC- und VAX-Anschluß

System-Familien:

Z80, 64180, 8085, NSC800, 65sc816, 6502, 6809, 68000, 68008, 80188/186

In Vorbereitung:

8088, 8086, 647180, Z280, 8051, 8052, 8051, 80535, 8048, 8049, 8035, 8039, Super-8

Ingenieurbüro für Datenverarbeitung
Dipl.-Ing. Hans-Jürgen Wilke
Adalbertsteinweg 26 D-5100 Aachen
Tel.: 0241/542228 FAX: 0241/533230

Leiterplatten-LAYOUT

Komplettes Leiterplatten-Entwicklungssystem für PCs:

- Schaltplan-Erstellung
- Interaktives Routen
- Auto-Routing
- 1/1000-Zoll Auflösung
- 1–8-Lagen-Multilayer
- 480×810-mm-Boards
- SMD-fähig
- Komfortable und schnelle Arbeitsweise
- Optimales Preis/Leistungsverhältnis

Demo (CGA/EGA, Mouse) 30.– DM

Ingenieurbüro für Datenverarbeitung
Dipl.-Ing. Hans-Jürgen Wilke
Adalbertsteinweg 26 D-5100 Aachen
Tel.: 0241/542228 FAX: 0241/533230

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

GWK GESELLSCHAFT FÜR TECHNISCHE ELEKTRONIK mbH

NEU
68020 – 68881
für
Selbstbaucomputer
c't 68000 GWK

Weltweit leistungsfähigstes Systemkonzept für modulare Systeme im Einfach-Europa-Format

Kostenloses Informationsmaterial anfordern!

Postfach 1360, D-5120 Herzogenrath,
Telefon (0 24 06) 60 35, Telex 8 32 109 gwk d

TRIO

Fakturierung, Mahnwesen und Buchhaltung

Das komplette Paket für Selbständige und Kleinunternehmen:

- 3 Funktionen in einem Programm, auf einer Disk
- einfache und sichere Benutzerführung, ausführliche Bedienungsanleitung (130 Seiten)
- **Interface für DATEV, dBase usw.**
- Artikelverwaltung, Kundenverwaltung, Serienbriefe, Umsatzstatistik, Rechnungsbuch, Creditoren, Debitoren, Kontenblätter, Betriebsübersicht, Journal, G&V, Bilanz
- max. 32 000 Kunden, Artikel und Buchungen
- **14 Tage frei zur Ansicht**

Der Buchhaltungsteil trägt das Prüfsiegel „anwenderfreundliche Software" und wurde von CHIP, C't, Softwaretestjahrbuch, PC+SOFT, hervorragend beurteilt.

780.– bis 980.– DM für XT/AT/PC

Informationen bei:

RÖNTGEN SOFTWARE
Oltmannsstr. 34, 7800 Freiburg, Tel. 07 61/40 87 40

ARCNET - NETZWERKE
2.5 Mbit / s

Starterkit für 2 Arbeitsplätze DM 2998,--

mit:
- Novell Netware ELS I
- 2 Netzwerkkarten
- 10 m Verbindungskabel
- Ausführliche Dokumentation

Einzelkomponenten:

Arcnet Netzwerkkarte für XT/AT	DM 450,--
Aktiver Hub (8-fach)	DM 998,--
Aktiver Hub (4-fach)	DM 480,--
Passiver Hub	DM 75,--
Netzwerkkarte für PS/2	auf Anfrage
Netzwerkkarten mit LWL	auf Anfrage

(alle Preise inkl. MwSt.)

Händler-Anfragen erwünscht !

Siebert Computersysteme
Jürgen Siebert
Wittelsbacherstrasse 80 * 5040 Brühl
Tel.: 02232 / 27644

Netware ist eingetragenes Warenzeichen von Novell Inc!
Arcnet ist eingetragenes Warenzeichen v. Datapoint Inc!

SRS63

Serieller ROM/RAM Simulator
mit 64 KByte

Die optimale Ergänzung für die Entwicklung von ROM-Anwendungen. Kein lästiges Epromprogrammieren in der Entwicklungsphase mehr.

Preis: 1368.– DM

FORTH-SYSTEME Postfach 1103 D-7814 Breisach am Rhein
ANGELIKA FLESCH Telefon (0 76 67) 5 51 Telefax 07667 555

Das kostenlose Christiani Kursprogramm

Lernen Sie heute, was morgen unentbehrlich ist.

Fordern Sie gleich das kostenlose Kursprogramm an, das Sie über unsere Lehrgänge informiert – u. a. auch über:

- ☐ Elektronik-Labor
- ☐ IC-Labor
- ☐ Digital-Labor
- ☐ Oszilloskop-Labor
- ☐ Fernseh-Labor
- ☐ Elektronisches Messen
- ☐ Mikroprozessortechnik
- ☐ Peripherie-Bausteine
- ☐ Kompakt-Kurs BASIC
- ☐ BASIC + Mikrocomputerpraxis
- ☐ Amateurfunk
- ☐ Automatisierung
- ☐ Optoelektronik-Labor
- ☐ El. Steuerungstechnik
- ☐ Videotechnik
- ☐ Kompakt-Kurs Elektronik
- ☐ CNC-Technik
- ☐ C64 CNC-Simulator
- ☐ Bautechnik
- ☐ EDV-Grundlagen

Gewünschtes ankreuzen; Anzeige ausschneiden und einsenden an

Dr.-Ing. P. Christiani · Technisches Lehrinstitut
Postf. 35 69160 · Tel. 0 75 31/58 01-0 · 7750 Konstanz

Industrie- u. Meßkarten für PC, XT, AT

– Copy-Board mit Laufwerkanschluß	229,–
– Bus-Extender-Karte f. PC, XT	98,–
– Colorgrafik- u. Herkules-Combi-Karte	298,–
– EGA-BAS-Adapter, wandelt RGB in SW-Video um	119,–
– IEEE-488-Karte zum Steuern für Meßgeräte	780,–
– RS 232 serielle Karte	89,–
– RS-422-Schnittstellenkarte	338,–
– Centronics-Karte für Druckeranschluß	89,–
– 35-MHz-AD/DA-Wandler, je 1 Kanal 8/10 Bit	548,–
– AD/DA, 12 Bit, 16 A/D- u. 1 D/A-Kanal	248,–
– AD-Wandlerkarte mit 8 Bit Auflösung, 1 Kanal	129,–
– Prototypen-Karte mit 24 TTL-I/O für XT, AT	198,–
– Relais-Karte mit 8255 u. 8 Relais m. Softw.	249,–
– Steppmotoren-Karte f. 2 Stepper m. Software	298,–
– Frequenzzähler von 300 Hz...1300 MHz f. PC/AT	298,–
– Digitalisierer m. 256 Graust. bis 1024 Pix.	991,–
– SSTV-Karte f. Bildübertragung per Funk m. AGC	328,–
– 48 I/Os mit 3x16-Bit-Zähler	248,–
– Optokoppler-Karte, 16 Eing./8 Ausgänge	368,–
– Eprommer 2716...27512 u. Textool-Sockel ext.	398,–
– PAL-Programmer für PC/XT/AT u. Software	798,–
– 640-KB-RAM-Karte, 0 KB best., für PC/XT	119,–
– 2-MB-RAM-Karte, 0 KB best., für PC/XT	329,–
– 3,5-MB-RAM-Karte, 0 KB best., für AT	339,–
– 4-MB-RAM-Karte, unbest.	389,–
– Modem-Karte f. Telefon, PC/XT, 300/1200 Bd. u. Software	339,–
– Game-I/O-Karte mit zwei Eingängen, kurze Karte	79,–
– Joystick für PC	36,–
– TVC-500, SW-Kamera mit Videoausgang, 220 V	448,–
– CCD-Industriekamera, 500 × 512 Pixel, ab 3 Lux	1799,–
– RS-232- auf RS-422-Wandler für lange Übertragungen	498,–
– Multiserielle Karte mit 4 ser. RS-232-Ports	229,–
– Serielle RS-232- u. Parallel-Printerport für XT u. AT	145,–
– 384-K-Multifunktions-Karte, 0 KB best.	189,–
– HDD/FDD-Controller, läuft bis zu 16 MHz	379,–
– Barcode-Pen mit vielen Normen u. Interfaces	850,–
– EPROM-Löschgerät für max. 6 EPROMs mit Timer	98,–

Bitte INFO anfordern. Händleranfragen erwünscht.
Versand erfolgt per NN. Auch Export möglich.

COMPUTER-RING
Heinrich Kolter Electronic, Postfach 11 27,
5042 Erftstadt-Lechenich, Steinstraße 22–24,
Telefon (0 22 35) 7 67 07, Telex 8 882 134

Konvertierungsprobleme?

lösen Sie mit der Hard- und Software von

A.S.S.-WARE

Mainframe *1) <–> MS-DOS
CP/M *2) <–> MS-DOS
MS-DOS *3) <–> MS-DOS

Die Konvertierungssoft- und Hardware läuft auf XT- und AT-kompatiblen Rechnern unter allen MS-DOS-Versionen ab DOS 2.X und 3.X.

*1) Z. B. Basisdatenaustauschformat: IBM 3540, 3601, 3602, 3740, 3741, (= IBM/370) 3742, 3747, 3962, 5110, 5114, 5231, 5320, 5340, 5380, Honeywell Bull, SEL usw.
IBM /32, /34, /36, /38. 8" und 5.25"-Disketten. Nixdorf 8870

*2) Fast alle CP/M-Formate von 3" bis 8"

*3) Fast alle MS-DOS-Formate von 160 KB bis 2 MB natürlich auch PS/2 1440 KB netto

Fordern Sie bitte ein Info an.

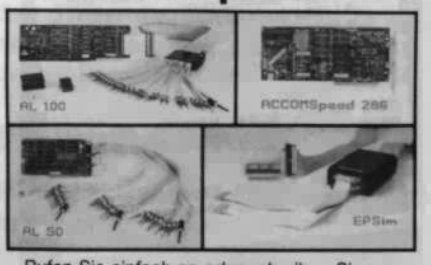
A.S.S.-WARE, Alfred Herrmann, Schimmelshahn,
5461 Roßbach/Wied, Tel. (0 26 38) 45 13

PC-ERWEITERUNGEN

DIE SICH JEDER LEISTEN KANN:

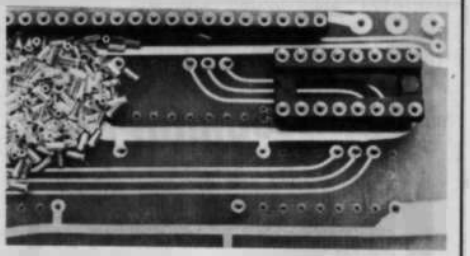
100-/25-MHz-LOGIK-ANALYSATOR AL 100
50-MHz-LOGIK-ANALYSATOR AL 50
8-/16-BIT-EPROM-SIMULATOR EPSim
Simuliert EPROMs von 16 bis 512 K
16-BIT-PROZESSORKARTE FÜR 8088-PCs

ACCOMSpeed 286



Rufen Sie einfach an oder schreiben Sie uns:

ACCOM System – Rudolf Berg und Werner Laub – GdbR
Röttererbergstr. 11A, D-7550 Rastatt, Telefon (0 72 22) 64 11



Durchkontaktieren

ohne Spezialwerkzeug mit
versilberten Kupferhohlnieten

Durchmesser (in Millimetern):							
außen	0,8	1,0	1,2	1,5	1,8	2,0	2,5
innen	0,6	0,7	0,8	1,0	1,2	1,5	2,0
Kragen	1,1	1,4	1,6	2,0	2,3	2,5	3,2
Menge:							
1 000	37.–	27.–	27.–	28.–	29.–	31.–	36.–
3 000	99.–	74.–	74.–	77.–	80.–	84.–	96.–
10 000	305.–	225.–	225.–	235.–	245.–	260.–	295.–

Rasterfolien, Kreppbänder, IC-Fassungen
Lieferung zzgl. Versandkosten per Nachnahme

Elmar Wienecke, Dipl.-Ing.
Wasserstraße 18 · 4973 Vlotho · Tel. (05733) 5801

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

MC BASIC-EMUF

Neu: mc-GAL-Programmer aus mc 2/88, programmiert GAL 16V8 und 20V8 mit BASIC-EMUF oder IBM-PC

Bausatz GAL-Programmer o. TEXTOOL	DM 98,–
Fertigplatine GAL-Programmer o. TEXTOOL	DM 148,–
3M-TEXTOOL-Sockel-Modul, 24polig	DM 49,–
Fertiggerät im Gehäuse für IBM-PC	DM 348,–
Basic-EMUF	
BS1: Platine, GAL, deutsches Manual	DM 98,–
BS2: BS1 + IP80C52, XTAL1	DM 198,–
BS3: RTC58321, AKKU	DM 29,–
BS4: µPD7002, ZN458, XTAL2	DM 29,–
BS5: ULN2003A = 75468, 2× Relais	DM 29,–
BS7: BS1 + TL7705, MAX232, IP80C52, 74HCT08, 74HCT373, 71055, RAM8K	DM 238,–
BSLCDKEY: Bausatz komplett	DM 198,–
FPLCDKEY: Fertigplatine komplett	DM 248,–
Basic-EMUF-Fertigplatine, komplett mit 32 kB RAM	DM 638,–
Basic-EMUF-Fertigplatine, Lochrasterversion mit 8 kB RAM	DM 438,–
PASCAL-CROSS-COMPILER (MS-DOS)	DM 969,–
CROSS-ASSEMBLER (MS-DOS)	DM 869,–
CROSS-SIMULATOR (MS-DOS)	DM 948,–

Intec Electronic GmbH
Rheingrafenstraße 37, 6501 Wörrstadt
Telefon 0 67 32/50 29

IBM-XT+AT-kompatibel

80386-Hi-Tech

ICO 386-1 4999.—
80386 CPU, 20 MHz. 1,2 MB-Lauf-werk, deutsche Tastatur, serieller, paralleler Port.

ICO 386-40 MB 5999.—
wie ICO 386-1 jedoch mit einer schnellen 38mS 40 Megabyte Festplatte und volle 2 MByte RAM. Höchste Rechenleistung für extreme Anforderung!

Maus 139.—
mechanisch MS-kompatibel an seriellen Port anzuschließen.

Barcodeleser 696.—
liest EAN, JAN, UPC, Codabar (NW7), 2 of 5 Interlave, Code 3 of9, Anschluß an Tastaturschnittstelle, dadurch keine Anpassungsprobleme

80286-Speed-Karte f. XT 339.—
AT-Power auf dem XT! 6/8 MHz

VGA-Karte 696.—
alle VGA Modes bis 640x480 (bei 16 aus 256 Farben) PS/2-Modell 30-kompatibel.

EGA-Karte (XT, AT) (640x350)	279.—
Super-EGA-Karte (640x480) Genoa kompatibel (XT, AT)	349.—
Super-EGA-Karte (800x600) Genoa kompatibel (XT, AT)	399.—
Seriell-Parallel-Karte (XT, AT)	99.—
22-MB Festplatte m. Controller u. Kabel f. XT	628.—
22-MB Festplatte ohne Controller	498.—
40-MB Festpl. (40 m/Sek.)	999.—
Monitor Grün (BAS-Anschluß für 12")	199.—
14 Zoll-Monitor (S/W 14")	299.—
EGA-Monitor (13")	898.—
Multisync-Monitor	1399.—

RATEV

Elektronik Vertriebs GmbH
Gothaer Straße 15, 4030 Ratingen 1
Telefon 0 21 02/4 20 51-52, Tx. 8 585 180 raf

286er-AT-Grundversion

Baby-AT-Board, XT-Format, Prozessor INTEL 80286, umschaltbar 6/8 (10) MHz, 0 Wait States, 1 MB RAM, bestückt mit 512 KB, Metall-Klapp-Gehäuse im Baby-AT-Format, drei Slim-line-Laufwerke möglich, Schlüsselschalter, LEDs für Power, Turbo Mode und Harddisk, Phoenix-BIOS, gepufferte Hardware-Uhr, 5× 16-Bit-Slot, 2× 8-Bit-Slot, Sockel für Co-Prozessor 80287, 180-W-Netzteil, parallele/serielle Schnittstelle (2. serielle optional), deutsche Tastatur mit 101 Tasten **DM 1850.–**

286er-AT-Standardversion

Grundgerät wie oben beschrieben, jedoch zusätzlich: Hercules-komp. Monochrome-Grafik-Karte mit zusätzlicher paralleler Schnittstelle, Western-Digital-Kombicontroller WD 1003 WA-2 (2× HD/2× FD), Mitsubishi-Floppylaufwerk MF 504 (360 K/1,2 MB) **DM 2498.–**

Zusätzliche Optionen

Aufpreise auf Standardversion:

20-MB-Festplatte, Fab. Seagate ST 225	DM 545.–
40-MB-Festplatte, Fab. Miniscribe MS 3650 (Slimline)	DM 980.–
EGA-Karte OEM 6, 640 × 350/480	DM 378.–
14"-Monitor, Fab. Addonics Dual Scan	DM 290.–
RAM-Aufrüstung auf 1 MB (512 K)	DM 160.–
Zweites Floppy-Laufwerk (1,2 MB)	DM 249.–
Zweites Floppy-Laufwerk (360 K)	DM 198.–
MS-DOS 3.2, deutsch, mit GW-Basic	DM 215.–

386er-AT-Grundversion

Baby-AT-Board, XT-Format, Prozessor INTEL 80386, 16-MHz-Takttfrequenz per Software umschaltbar, 0 Wait States, 2 MB RAM bestückt, bis 8 MB erweiterbar, 4× 8-Bit-Slot, 4× 16-Bit-Slot, ansonsten wie 286er-Grundversion **DM 4500.–**

286er-Baby-AT-Mainboard wie oben beschrieben	DM 998.–
386er-Baby-AT-Mainboard wie oben beschrieben	DM 3800.–
386er-Baby-AT-Mainboard, 20 MHz, 32 Bit geprüfter INTEL-Prozessor, 2 MB bestückt	DM 4499.–

Chipsätze von CHIPS + Technology

ICO AT-1 1799.—
80286-CPU, 6/8/10 MHz, 640 K RAM, Color-Karte, 1,2 MB-Disk, deutsche Tastatur.

ICO AT-20 2599.—
Der Standard-AT: AT-1 m. 20 Megabyte Festpl.

ICO AT-30 2699.—
Wie AT-1 jedoch mit 30 MB-RLL-Platte.

12-MHz-Speed + 1MByte 292.—
(statt der 640 KB) für obige ATs. Die 12 MHz vertragen sich durch neuartige Tastanpassungsschaltung. (DSC) auch mit Zusatzkarte.

AT-Konfigurationen

statt Color-, eine Monokarte	+10.-
statt Color-, eine EGA-Karte	+180.-
extra serieller + parallele Schnittstelle	+99.-

ICO 360 888.—
IBM XT kompatibler Rechner mit 8088 CPU 256 KB RAM, 360 KByte Laufwerk, Colorkarte, deutsche Tastatur, im stabilen Metallgehäuse im original-look.

ICO 20MB 1499.—
Der XT-Profi. Wie ICO 360 jedoch mit einer 20 MegaByte Festplatte eingebaut.

Andere XT-Konfigurationen

Monochromkarte statt Colorkarte	+10.-
Extra Drucker-, Serielle-, Game-Port, Uhr	+70.-
8 MHz-Fastspeed (statt 4,77 MHz, original)	+40.-
Extra 2. Laufwerk	+249.-

Bondwell BW 8 1599.—
8088 CPU, 512 KB-RAM, LCD-Displ, mit 640x200 Punkte Grafik, 3,5" Disk, Uhr, serieller Port, Druckeranschluß u. Anschl. f. 2. Laufwerk, Akku-Betrieb. Incl. MS-DOS u. GW-Basic Gewicht 4,5 kg.

5¼ Zoll Diskdrive zu BW 8 499.—

BW 8 S 1699.—
(Supertwist) Mit Supertwist-Display, bsd. hoher Kontrast

7 Monate Garantie Versand erfolgt per NN. Händler: Bitte günstige Händlerpreisl. anfordern.

Klaus Jeschke
Hard-, Software
Adelheidstr. 2-20
6240 Königstein
☎ (06174) 3041

SHARP PC 1600

Wir haben d i e Lösung für Ihre Speicherplatzprobleme.

P C - 1 6 0 0

Unsere Speichererweiterungsmodule werden einfach in Slot 2 bzw. Slot 1 (32 K-Module) geschoben.

RAM-MODULE	normal	extra**
32 K-Byte	159,00 DM	209,00 DM
128 K-Byte	489,00 DM	529,00 DM
256 K-Byte	639,00 DM	689,00 DM

extra** = wahlweise akku- oder batteriegepuffert mit Schreibschutzschalter

E-PROM
32 K-Byte 110,00 DM / Programmieradapter 120,00 DM

P C - 1 5 0 0

Interner Speicherausbau: Grundpreis 136,80 DM
plus 22,50 DM pro 8 K-Byte. Bis 568 K-Byte.

Porto/Verpackung zu Selbstkosten. Wir gewähren ein Jahr Garantie.

Achtung !!, Neue Anschrift !! Achtung !!
E L K E B A J I Ć
Planungsbüro
Hardware · Software · Elektrotechnik
Bauer Landstraße 99 2390 Flensburg
Telefon: (0461)420-37/38 Telefax: (0461)45026

mc-AllProg Plus

Universelles Programmiergerät aus mc 6-8/1987 mit erweiterter Software

+ Programmiert jetzt auch GALs.
PROMs von TI und Zeropower-RAMs
+ Vorbereitet für Single-Chip-Prozessoren von INTEL
+ Unterstützt das INTEL-Hex-Format
+ Inhaltsverzeichnisanzeige

* Programmiert (E)EPROMs bis 27512, 27513 und 27011. PROMs von Signetics/Valvo und PALs von MMI, NS, TI und AMD
* Anschluß über RS 232 (V.24) Schnittstelle
* Bediensoftware für IBM-PC/XT/AT (unter PC/MS-DOS), kompatible und Z80-Systeme (unter CP/M) verfügbar
* Bildschirmorientierter (E)PROM-Editor
* Bildschirmorientierter Text-Editor zum PAL-Entwurf
* PAL-Assembler + PAL-Simulator + PAL-Disassembler
* Fuselisten als JEDEC-Dateien speicher-/lesbar

\# Bausatz (inkl. Handbuch und Software) ..... 997,– DM
\# Fertigplatine (getestet, Handb., Software) .. 1420,– DM

Versand per Nachnahme. Bei Bestellung unbedingt Diskettengröße, Diskettenformat und Betriebssystem angeben!

Holger Haase & Michael Menrad
Peripheriegeräte GbR
Postfach 2647 / 3300 Braunschweig
Telefon: (0531) 400310

ELINK-03 Datenfernübertragung ohne Probleme

- 300 BAUD vollduplex
- 1200 BAUD halbduplex
- 1200/75 BAUD vollduplex
- 75/1200 BAUD vollduplex
- ELINK-03 wählt vollautomatisch und nimmt Anrufe entgegen (MAILBOX)
- SMARTMODEM kompatibel: arbeitet mit Standardsoftware wie OPEN-ACCESS, SYMPHONY, KERMIT, CROSSTALK usw.
- Eingebaute Intelligenz erledigt Datentransfer mit jedem Computer oder Terminal über V-24-Schnittstelle, natürlich mit automatischer Baudratenerkennung
- Einfache Programmierung für Spezialaufgaben durch den Anwender
- Betrieb an Nebenstellenanlagen möglich
- Postzulassung mit FTZ-Nummer
- günstige Mengenstaffeln
- komfortables MAILBOX-Programm ebenfalls erhältlich

MICOM Computer- und Informationssysteme Entwicklungs- und Vertriebsges. mbH
Wüstenhofer Str. 6 · 5600 Wuppertal 1 · Tel. 02 02/44 34 01

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

C-COMPILER

MI-C für CP/M, CP/M 86, MS DOS

Nutzen Sie die Vorteile von C
MI-C vereint hohen Bedienungskomfort mit hervorragender Leistung

- Vollständige Version mit 13stelliger BCD-Arithmetik für Gleitkommazahlen
- Erzeugt kurze und schnelle Programme, die auch in ein ROM gebraucht werden können
- Ausgabe in 8086- oder Z80-, 8080-Assemblercode
- Kompatibel zu MAC80/L80 (MASM/LINK) v. Microsoft
- Für 8086: 4 Speichermodelle/8087-Co-Prozessor
- Fehlerverfolgung mittels Trace möglich
- Umfangreiche Bibliothek (AMD9511-Paket erhältlich)
- UNIX-kompatibel
- Deutsche oder englische Version lieferbar 8"-/5,25"-/3,5"-/3"-Disk + dt. Handbuch

MI-C für CP/M	445.– DM
MI-C für CP/M 86 oder MS-DOS	575.– DM
MI-C Cross (Zielprozessor 8080/Z80)	745.– DM
MI-C Crossassembler + Linker (Ziel 8080/Z80)	645.– DM
MI-C Crosspaket (Ziel 8051)	1495.– DM

Herbert Rose, Bogenstraße 32, 4390 Gladbeck, Telefon 0 20 43/2 49 12 und 4 35 97
Vertrieb in Österreich:
Dr. Willibald Kraml, Microcomputer-Software, Degengasse 27/16, A-1160 Wien

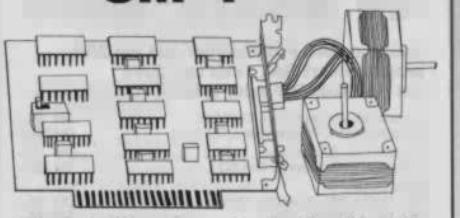
Der Eprommer für **IBM** und -kompatible **Apple** //e, //gs,][+, und **CPC 464/664/6128**

✱ Programmiert alle gängigen EPROM- und EEPROM-Typen (z. B. 2716, 27C16, 2732, 2732A, 27C32, 2758, 2764, 2764A, 27C64, 27128, 27128A, 27C128, 27256, 27C256, 2508, 2516, 2532, 2564, X2804A, X2816A, X2864A ...) ✱ Menügesteuerte Software auf Diskette/Kassette ✱ 32 KByte frei für EPROM-Daten (Brennen des 27256 ohne Nachladen) ✱ Kein Umschalten, Stecken oder Löten nötig ✱ Programmierspannungen werden im Gerät erzeugt ✱ Verbindung zum Rechner über Flachbandkabel ✱ Rote und grüne LED zur Betriebsart-Anzeige ✱ Komplett mit 28pol. Textool-Sockel ✱ IBM- und CPC-Version mit Interface-Karte (durchgeführter Expansionsport bei CPC-Version) ✱

Preise für IBM: Fertiggerät **DM 399.50** ✱ Bausatz **DM 349.–**
für Apple: Fertiggerät **DM 269.50** ✱ Bausatz **DM 219.–**
für CPC 464/664: Fertiggerät **DM 289.50** ✱ Bausatz **DM 239.–**
für CPC 6128: Fertiggerät **DM 319.50** ✱ Bausatz **DM 269.–**
CPC-Software auf 3"-Diskette statt Kassette: + DM 15.–

DOBBERTIN
INDUSTRIE-ELEKTRONIK **GmbH**
Brahmsstraße 9, 6835 Brühl, Tel. (0 62 02) 7 14 17

SM-1



Die Schrittmotorenkarte für XT, AT

- 2 bipolare Schrittmotoren anschließbar
- 8-Bit-Eingabeport (TTL-Pegel)
- 6-Bit-Ausgabeport (o. K. 500 mA/50 V)
- Motorspannung intern/extern umschaltbar
- Stromreduzierung für Ruhezustand

Steckkarte aufgebaut u. getestet mit 2 Motoren 3,6°/1,8° 12 V **DM 189,–**
Versand per Nachnahme

Soft- und Hardware Tetzlaff
Grotenbachstraße 9, 4600 Dortmund 50
Telefon 02 31/71 30 42

HIGH-SPEED MESSDATENERFASSUNG

DAS OSZILLOSKOP IM PC: (für XT/AT kompatible)

PC-SCOPE T3240: 4900 DM (5586,00 inkl.MwSt)
2 Channel 40MHz 8Bit 20/40MHz Sample Rate
16/32 KByte RAM 5mV...20V/DIV Basis-Software

PC-SCOPE T1620: 2980 DM (3397,20 inkl.MwSt)
2 Channel 8MHz 8Bit 10/20MHz Sample Rate
8/16 KByte RAM 7,5mV..15V/DIV Basis-Software

PC-SCOPE T0410: 1240 DM (1413,60 inkl. MwSt)
1 Channel 10MHz 8Bit 10MHz Sample Rate Basis-Softw.

DER FUNKTIONSGENERATOR IM PC

PC-SIGNAL S3210: 1950 DM (2230,00 inkl.MwSt)
2 Channel 12Bit 2x16 KByte RAM 10 MHz Clock Software

KOMPLETT-SYSTEME:

286 PORTABLE + 2xT3240 – 4-Channel Transienten-recorder incl. Software 15900 DM (18126,00 inkl.MwSt)

19 Zoll 286-AT + 4xT3240 – 8-Channel Transienten-recorder incl. Software 21500 DM (24510,00 inkl.MwSt)

optionale Software zur Signalanalyse (FFT, X/Y-Darstellung, Kanalverknüpfung) und Dokumentation usw. auf Anfrage

intelligente Meßsysteme GmbH

7150 Backnang, Uhlandstraße 16, Tel 07191-63042 btx
8031 Eichenau, Hauptstraße 17, Tel. 0 81 41-8 20 22

ALMIK

Mindestbestellmenge je Ausführung 10 Stück

DM **1.29**

Fordern Sie unsere Preisliste an

D-Sub-Steckverbinder
mit gedrehten Präzisionskontakten, hartvergoldet über Nickel, in allen Polzahlen jederzeit ab Lager lieferbar, direkt aus eigener Fertigung, zum

Sub-Miniatur-Preis

z.B. 25polige Stiftleiste für Handlöten .... St. **DM 1.29**
z.B. 25polige Buchsenleiste für Handlöten St. **DM 1.90**
zzgl. 14 % MwSt., 1.70 DM Nachnahmegebühr u. Porto.

Almik, Wolfratshauser Straße 27, 8000 München 70
Tel. (089) 7241075 (Sammelanschl.), Telex 529920 almik d

Preis-Sensation

DIN-A3-Plotter
mit Papierbewegung
TSS 860

HP-GL-kompatibel

DM 3398,–

6 Farben
0,025 mm Auflösung
400 mm/s Zeichengeschwindigkeit
Centronics- und V.24-Interface
56 Zeichenbefehle

Lieferung per Nachnahme

TSS-Schmitz
In der Holl
5223 Bierenbachtal · Tel. 0 22 93/21 88
Inh. Brigitta Schmitz

mc 3/88

ZECK FOR 386 TOWER CASE AND SWITCHING POWER SUPPLY

Produkte für IBM-PC/AT Kompatible:

ZKS-820 200W
ZKS-818 180W
ZKS-820L 200W
ZKS-818L 180W

ZKS-220 220W

ZKS-202 200W
ZKS-201 200W
ZKS-201A 200W
ZKS-203 220W
ZKS-TWR-3

U.P.S. ZPS-500 * Distributor Wanted

ZECKS ENTERPRISE CO., LTD.
No. 38,35 LANE SEC 6 CHUNG SHAN N RD. TAIPEI
TAIWAN R.O.C. FAX: 886-2-8311744 TEL: 886-2-8342525
TLX: 29266 ZECK

HARDWARE-MESSWERTERFASSUNG

– IBM XT/AT – CBM – hier einige Auszüge IBM –

IEEE-488-(IEC-Bus-)Platine und Software	ab DM 480
32-Bit-Optokoppler-Input-Platine	DM 480
32-Bit-Optokoppler-Output-Platine	DM 480
8-Bit-32-Kanal-A/D-Wandler, 8 us Wandlung	DM 990
8-Bit-32-Kanal-A/D-Wandler mit RS-232-Schnittstelle	DM 990
12-Bit-32-Kanal-A/D-Wandler, 25 us, Uni-/Bipolar	DM 990
12-Bit-4-Kanal-D/A-Wandler, 2 us	DM 350
72-Bit-Input-/Output-Platine, 8/4 bitweise programmierbar	DM 540
192-Bit-Input-/Output-Platine, 8/4 bitweise programmierbar, gesockelt	DM 760
Relais + I/O-Platine (12 Relais 1 x un. 220 VAC, 3 A/60 W + 12 x TTL-I/O)	DM 1675
8fach-RS-232-Umschaltplatine (Umschaltung programmierbar od. Alt+1...8)	DM 990
Multiboard 86, 12 Bit–32 K, A/D + 12 Bit–2 K, D/A + 96 TTL-I/O, gesockelt	DM 990
Multiboard 87, 8 Bit–32 K, A/D + 8 Bit–4 K, D/A + 80 TTL-I/O	DM 990
Thermoboard 86, -50 °C–+150 °C, 16 Kanäle, od. d. -50 °C–+1150 °C	ab DM 990
Centronics-/IEC-Interface f. Drucker, Plotter mit IEC/IEEE-488)	ab DM 295
Programmierbarer Timer-Counter, 8fach/24fach, 16 Bit	DM 350/DM 660

NEU IM PROGRAMM

Elektrostatische Plotterauflagefolie zum Festhalten des Plotterpapiers	DM 295
16-Bit-A/D-Turboboard, 16 us, mit 128 KByte RAM	DM 1650
12-Bit-A/D-Turboboard, 25 us, mit 128 KByte RAM	DM 1290
Logic-Analyzer-Card, 30 MHz/50 MHz/100 MHz	DM 980 – DM 1453 – DM 2137
32-Bit-Open-Kollector-Output-Platine, max. 50 V/500 mA	DM 390
PT-100-Verstärker für versch. Bereiche, 100 °C, 200 °C...800 °C	DM 150
16-Bit-8-Kanal-A/D-Wandler (Dual-Slope)	DM 920
8048-Emulator-Kit inkl. 2-Pass-Assembler	DM 1190
Einsteilbare Quarzzeitbasis	DM 280

Info kostenlos! **LOTHAR BOCKSTALLER**
Hard- und Software – Hadwigstr. 16, 7867 Wehr 2, Telefon 0 77 61/18 08

Preis-Sensation

mc 3/88

DIN-A3-Plotter
mit Papierbewegung nur DM **1990,–**

HP-GL-kompatibel, 4 Farben
Serielle und parallele Schnittstelle
Lieferung per Nachnahme

TSS-Schmitz, Inh. Brigitta Schmitz
In der Holl, 5223 Bierenbachtal
Telefon (0 22 93) 21 88

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

PLATINEN-LAYOUT

Leiterplatten-Entwurf auf PC/XT/AT-Systemen

BOARDSTAR ist ein Software-Paket für Entwurf und Reinzeichnung gedruckter Schaltungen und bietet Ihnen professionelle Leistungsmerkmale zu einem fairen Preis. Komfortable mausgesteuerte Benutzeroberfläche - Leiterbahnführung im 1/40" Raster - Bearbeitung doppelseitiger Platinen bis 516 qcm - Verwaltung von Macro-Bibliotheken - Reinzeichnung mit jedem handelsüblichen Plotter.

ROUTSTAR ist der Autorouter zum BOARDSTAR und erstellt selbsttätig Layouts aus vorgegebenen Schaltungsbeschreibungen: Layout-Beschreibung durch Grafik- oder Textfiles - Berücksichtigung vorgegebener Designkriterien - Problemlose Verarbeitung von BUS-Strukturen.

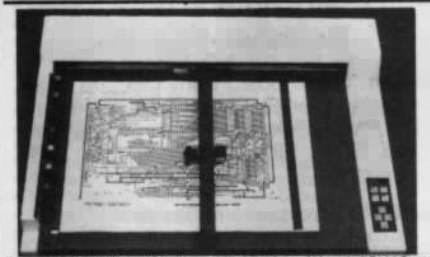
BOARDSTAR (interaktiver Layout-Editor)	698,- DM
ROUTSTAR (Autorouter zum Layout-Editor)	698,- DM
BOARDSTAR-Paket (enthält BOARDSTAR+ROUTSTAR)	998,- DM
Lauffähiges Demo-Paket (wird angerechnet)	70,- DM
DIN-A3 Flachbett-Plotter PL-450 (6-Stift/HPGL/0,025mm)	2450,- DM

Wir bieten Ihnen auch ein breites Spektrum von Service-Leistungen rund um das BOARDSTAR-Programm, vom Plot- u. Repro-Service bis zur Platinenfertigung nach angelieferten BOARDSTAR-Disketten. Fordern Sie Informationsmaterial oder besser gleich das Demopaket an.

Datentechnik Dr.Gert Müller Diezstraße 2A
D-5300 Bonn 1
Tel. 0228/217297

TURBO-PLOTTER

Schnell, preiswert und präzise plotten mit PL-450



Format bis DIN-A3, HP-GL* kompatibel, serienmäßig mit RS232C und Centronics-Schnittstelle ausgerüstet, Pen-Speed maximal 565 mm/s, programmierbarer Stiftwechsel für 8 Farben (Pens), Aufnahme für HP-Standard Faser-/Keramik-/Tusche-Pens, Pen-Magazin mit Automatic-Verschluß, Auflösung 0,025 mm, verschied. Zeichensätze und und.....

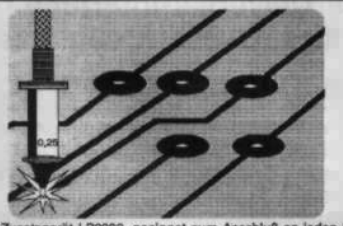
Unser Preis: ..... 2450,00 DM

Datentechnik Dr.Gert Müller Diezstraße 2A
D-5300 Bonn 1
Tel. 0228/217297

* HP-GL ist ein Warenzeichen der Hewlett-Packard GmbH

FOTOPLOTTER

Wir machen Ihren Penplotter zum Fotoplotter!



Fotoplot-Zusatzgerät LP2002, geeignet zum Anschluß an jeden Flachbett-Plotter. Gleichbleibende Schärfe und Strichbreite durch geschwindigkeitsabhängige Steuerung der Lichtintensität. Hervorragend geeignet zur Herstellung von Leiterplatten-Filmen mit CAD-Systemen.

Fotoplot-Zusatz LP2002 .....2223,00 DM

Fordern Sie Informationsmaterial an!

Datentechnik Dr.Gert Müller Diezstraße 2A
D-5300 Bonn 1
Tel. 0228/217297

AZTEC C - COMPILER MANX

- Native-Compiler für MS-DOS, CP/M80, CP/M86, Apple, Amiga
- Cross-Compiler für 80186/286, Z80/8085, 6502 und 68000
- Compiler, Assembler, Linker, Hex-Converter, HLL-Debugger
- alle Tools zur Eprom-Erzeugung sind enthalten

Schema II Schaltplan - CAD

- superschneller Bildaufbau mit Zoom und Scrolling
- DESIGN-RULE-CHECK, Stücklisten- und Verbindungslisten
- Bauteile - Bibliotheken für Elektronik und E - Technik

PROTEL-PCB Platinen - Layout

SPITZENLEISTUNG ZUM SUPERPREIS !!!

- manueller Router mit 1/1000 Zoll Auflösung
- unterstützt Multilayer, SMDs, Gerberplots und Excellon
- automatische Netzlistenkontrolle mit Soll/Ist - Vergleich
- interaktiver Autorouter als Erweiterungspaket

Demodisketten für SCHEMA und PROTEL je DM 50,--

INGENIEURBÜRO DIPL.ING. MANFRED SUCHY
Gottlieb-Daimler-Str. 12 8037 Olching
Telefon 08142/12360 und 08142/28028

Pocketmodem 1200 498,–

Nur 10 × 6 × 2,4 cm klein, für Portabels usw.

Modemkarte PC 1200 349,–

Kurze PC-Karte, V21/22, 300/1200 Bd vollduplex, RS 232 eingebaut (COM 1–4), Lautsprecher.

KS 1200, extern, sonst wie PC 1200 ..... **398,–**
KS 2400, 2400/1200 Bd vollduplex **598,–**
Multimodem, V21/22/23 ..... **598.–**
Alle Modems Hayes-komp. und **ohne FTZ-Nr.**

RAMboard 2 MB 349.–

EMS-/Intel-komp., ohne RAM, mit Software
AT-RAMboard mit 16-Bit Slot, sonst w. o. **449,–**

AD/DA-Wandler 269,–

Auflösung 12 Bit, 16 AD-, 1 DA-Kanal

VIA 8255 179,–

I/O-Karte, 48 Leitungen, 16 LED, Timer/Counter

ARClink-Netzwerkkarte 790,–

Z. B. für Novell, Info über weitere ARClink-Produkte anfordern

bsb

D. Schöddert – B.v.d. Brincken
Hohenzollernring 74
5000 Köln 1
Telefon **02 21-13 14 41**

Datentechnik

EPROM-Löscher

Kurze Löschzeit (10 min)

1000fach bewährt

ELG-3, batteriebetr. Löschlampe für 3 EPROMs gleichzeitig, ohne Batterien ..... DM 80.–

ELG-4, netzbetr. Löschgerät mit Schaltuhr und Schublade für 7 EPROMs gleichzeitig DM 180.–

Löschset, bestehend aus: UV-Röhre TUV 4 W, Vorschaltgerät, Starter, Fassungen, Schaltuhr und Schaltplan ..... DM 58.–

BIBUG-8, Realtime-Debugger mit den Modulen für: Z80, 8085, 6502, 6510, 6800, 6802, 6809 u. ä., LCD-Anzeige des Adreß- u. Datenbusses, Realtime-Test oder Single-Step. Einfacher Anschluß über CPU-Sockel ohne zusätzliche Stromversorgung. Preis inkl. einem Modul ..... DM 956.– weitere Module ... DM 106.–

Hard- und Software-Tester

Entwicklungsbüro Fritz Krickl
Schauinslandweg 27, 7730 VS-Schwenningen,
Telefon (07721) 71442

SOFTWARE & HARDWARE

für Apple II & Mac & MS-DOS-Computer

z. B. für APPLE II und Kompatible

Wir liefern die RAMworks-Karte (AE) für den Apple IIe mit max. 16 MB (z. B. erweitert Appleworks) z. B. RAMWORKS III 64 K ..... DM 469.–
Anpassung für Appleworks 1.2 auf dem II+/e.
Original oder mit externer Tastatur. Anpassung in deutsch für SATURN 128 K und IBS AP33 1 MB! ..... DM 130.–
AD 16 Ch. 12 Bit, schnell! (Applied Eng.) ..... DM 799.–
PKASO/U-Printer-Karte (IS) ..... DM 429.–
Timemaster II H. O., die Uhrenkarte! (AE) ..... DM 299.–
ELF kompl. Statistik-Software (TWG) ..... DM 495.–
Prime-Plotter-Grafik-Software (Primesoft) ..... DM 629.–
Z-RAM Ultra 3, 256 K, Z80 CP/M + Uhr, max. 1 MB ..... DM 798.–
Z-RAM Ultra 2, 256 K, mit Uhr, max. 1 MB ..... DM 698.–

z. B. für IBM und Kompatible

MatchPoint-PC plus UNIFORM-PC (Lesen/Schreiben von Apple Disks im IBM PC/XT/AT/100 % komp. für DOS-ProDOS-CP/M) ..... DM 649.–
MatchMaker (Anschluß von Mac 800 K Drive an IBM PC/XT/AT/100 % komp. zum Datentransfer Mac-MSDOS) ..... DM 498.–
CompatiCard (Anschluß von 3.5", 5.25", 8" Drives an IBM PC/XT/AT/100 % komp. z. B. 3.5" 720 K/1.44 MB) ..... DM 629.–
XENO-PC (Cisin) (Lesen/Schreiben von CP/M- u. MS-DOS-Formaten im IBM PC/XT/AT/100 % komp.) ..... DM 258.–
ELF PC kompl. Statistik Software (TWG) ..... DM 500.–

z. B. für alle Systeme

Printerchanger 3 parall. Drucker auf 1 Micro inkl. Kabel/Netzteil (Keyzone) ..... DM 340.–
Printersharer 3 Micros auf 1 parall. Drucker inkl. Kabel/Netzteil (Keyzone) ..... DM 340.–

Wir sind Import-Spezialisten und bieten Ihnen eine große Auswahl an Software und Hardware bedeutender Hersteller aus den USA und England. Informationen gegen DM 3.– in Briefmarken.

WEISS COMPUTER Dipl.-Psych. Karl-Heinz Weiß
Am Wiesenhof 17, 2940 Wilhelmshaven, Tel. 0 44 21/8 31 79

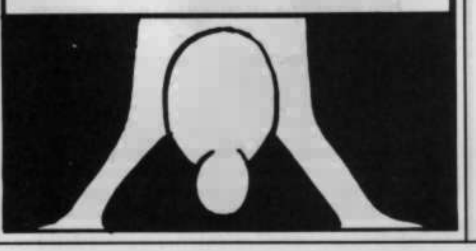
Berliner Straße 73
Postfach 250166
5090 Leverkusen
Tel. (0214) 95060

moVe

TELETEX mit dem PC

TELETEX mit deutschem Know-how:
mbp UTC 101, inkl. deutscher Software und deutschem Handbuch sowie Montage **(bundesweit)** DM 5398.–

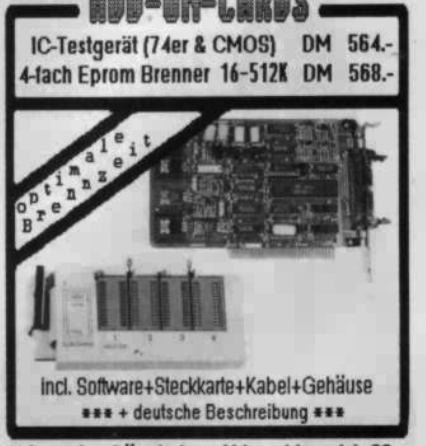
Bitte fordern Sie unsere detaillierten Unterlagen an.



IBM PC/AT und kompatible Zusatzkarten:

ADD-ON-CARDS

IC-Testgerät (74er & CMOS) DM 564.-
4-fach Eprom Brenner 16-512K DM 568.-



incl. Software+Steckkarte+Kabel+Gehäuse
***** + deutsche Beschreibung *****

Ingenieurbüro Lutger Ahlers, Mozartstr.23
8052 Moosburg, Tel. 08761/4245

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

IEC-BOX

Universal-Interface für IEC-Bus-Meßplätze

- Die preisgünstige Erweiterung für alle IEC-Systeme als Front-End für Meßdatenerfassung, Automatisierung und Meßtechnik
- Durch Einschubkarten flexibel erweiterbar
- Mikroprozessorgesteuert (6502/65802)
- Anschließbar auch über V.24 oder Current-Loop
- Ansprechbar aus allen Programmiersprachen

Grundgerät ..... DM 1495,–
+ ges. MwSt.

ees GmbH, Pelkovenstraße 51
8000 München 50
Telefon 0 89/1 41 10 77

6502/65816

Europakarten-Rechner-Familie

- Vom Single-Board-Computer bis zum vollständigen Steuerrechner mit kompletter Peripheriekarten-Palette (digital, analog, Grafik, Bildverarbeitung, IEC, V.24, Prommer)
- 16 MB adressierbarer Speicher
- Festplatte bis 80 MB
- Programmiersprachen:
  - BASIC
  - FORTH
  - PASCAL
  - ASSEMBLER

ees GmbH, Pelkovenstraße 51
8000 München 50
Telefon 0 89/1 41 10 77

Hermann-Josef Miele Datentechnik GmbH
Bergfreiheit 60
5788 Winterberg-Silbach
Tel. (0 29 83) 83 07 u. 83 37

Modula 2 für 68000-Systeme

OS-9-Version Preis ab DM 1368.–

UNIX-Version Preis ab DM 2850.–
für HP9000 300er Serie, Convergent Technologies, NCR, Olivetti u. v. a.

Des weiteren haben wir ein Riesenangebot an Software für OS-9 und UNIX.
Bitte fordern Sie unseren Katalog an.

Mehrplatz-Systeme mit OS-9 oder UNIX VMEbus oder ECBbus

mtr3, DER EPROM-HELFER

sekundenschnell — automatisch — handlich

- für EPROMs und EEPROMs bis 256Kbit (Option 1Mbit),
- Adapter selbst für wenig gebräuchliche Bauelemente,
- sekundenschnelle aber auch konventionelle Programmierung,
- Typwahl automatisch oder zumindest ohne Datenblatt-Hilfe,
- Emulatorfunktion
- V.24/RS232C-Schnittstelle, einfach konfigurierbar,
- Netz- und Akkubetrieb,

und das alles im kleinsten (189 x 138 x 48) autonomen EPROM-Programmierer für

DM 2257,20 Sfr 1950,— exkl. Wust

Meßtechnik Dr.-Ing. Ranfft
Dörpfeldstr. 15, D-5657 Haan 2, Tel. 0 21 04/6 28 27

Hedinger + Partner KG
Zentralstr. 3, CH-8610 Uster, Tel. 01/940 90 55

EDV-Systeme und Zubehör

BTX + TELEX

Blaupunkt-BTX-Decoder
für PC/XT/AT DM 1140.–

BTX-Decoder + Profi-TELEX-Programmpaket zum automatischen Telexempfang und -versand DM 1480.–

Unser weiteres Lieferprogramm:
Festplatten-Subsysteme von 20 MB bis 260 MB
Erweiterungskarten für PC/XT/AT
Modems – Kabel – AST-Produkte – Monitore
WCT-Personal-Computersysteme
und vieles mehr

Wir stellen aus: Cebit 88 Halle 4, EG Stand E 60

WILHELM Computertechnik GmbH

Niederstr. 14 · 4019 Monheim · Tel. 0 21 73/5 00 25
Telefax 0 21 73/3 14 53 · Btx 0 21 73/5 00 25

ADS-Industrie- + -Meßkarten für XT, AT

Multifunktionskarte, 64 × A/D, 2 × D/A, 24 × I/O	DM 325
Ein-/Ausgabekarte, 48 × I/O, über 8255,	
3–16-Bit-Zähler	DM 164
Ein-/Ausgabekarte, 16 Optokoppler, 16 Relais	DM 427
12-Bit-Wandlerkarte, 16 Kanäle A/D, 1 Kanal D/A	DM 247
14-Bit-Wandlerkarte, 16 Kanäle A/D, 1 Kanal D/A	DM 407
12-Bit-Wandlerkarte Dual-Slope (0–4 V oder 4–20 mA)	DM 539
Prototypenkarte inkl. Decoder	DM 190
D/A-Wandlerkarte, 8 Kanäle/8 Bit	DM 553
D/A-Wandlerkarte, 8 Kanäle/12 Bit	DM 892
Prototypenkarten	ab DM 68
Alle Meßkarten mit deutscher Beschreibung und Programmbeispielen.	
IEEE-Karte mit Basic (Pascal opt.)	DM 480
PAL-Programmierer inkl. Software	DM 620
Epromprogrammierer bis 1 MBit	ab DM 335
Logic-Analyzer, 100 MHz	DM 1980

Fordern Sie unseren kostenlosen ADS-Katalog an. Wir haben 36 verschiedene Interface-Karten auf Lager!

Niedermeier Computer Products GmbH
Allmannsberg 1, 8094 Edling, Tel. 0 80 39/12 95

LAYTRONIC

PCB-FULL-SERVICE

- *Film-plots für alle CAD- und Low-cost-CAD-Systeme im HP/GL-, Gerber- und Marconi-Format, in 24 Stunden!*
- *Plot-Service mit Muster-prints im Eildienst*
- *Klein- und Mittelserien*

LAYTRONIC GmbH · Industriestr. 7, 7434 Riederich
Telefon 0 71 23/3 36 35, 3 42 23, Telex 7 245 409

LAYTRONIC

Aus eigener Fertigung:

- *Musterleiterplatten-Schnelldienst*
- *Klein- und Mittelserien*
- *Multilayer*
- *Auf Wunsch elektrisch geprüft*
- *Kurze Lieferzeiten durch rationelle Fertigungsmethoden*

LAYTRONIC GmbH · Industriestr. 7, 7434 Riederich
Telefon 0 71 23/3 36 35, Telex 7 245 409

DIN-A3-Plotter PLX-88

Spitzentechnologie zum Sensationspreis

Made in Germany

Preis DM **1498,-**

- DIN-A3-Plotter
- 4 Farben m. autom. Stiftwechsel
- 0,025 mm Auflösung
- max. 200 mm/s Zeichengesch.
- HPGL-Kompatibel 28 Befehle
- Serienmäßig zwei Schnittst.
  Centronic u. RS 232 (V-24) ohne Aufpreis
- Ladbare Zeichensätze

CMC GmbH
Gesellschaft f. Electronic u. Microprozessorsysteme mbH,
Sebastianstraße 5, D-8480 Weiden, Telefon 0961/35053

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

MINIPREISE FÜR LAUFWERKE

PHILIPS	X3132 2 x 40 Spur slim line	DM	209,–
PHILIPS	X3134 2 x 80 Spur slim line	DM	240,–
	mit Umschaltung 40/80 Spur	DM	255,–
PHILIPS	X3113 1 x 80 Spur 2/3 Bauhöhe	DM	95,–
	mit Umschaltung 40/80 Spur	DM	110,–
PHILIPS	X3114 2 x 80 Spur 2/3 Bauhöhe	DM	209,–
	mit Umschaltung 40/80 Spur	DM	229,–
Floppygehäuse für slim line		DM	25,–
Tastaturen ohne Gehäuse		DM	25,50
Datenkabel für 2 Laufwerke		DM	32,–
Anschlußstecker für Stromversorgung		DM	2,95

Alle Preise zuzg. Versandkosten, Versand per NN oder Vorkasse

CHRISTEL VON DER LINDEN 4200 OBERHAUSEN
STERRADER STR. 189 TEL. 0208/663721 AB 14 UHR

V.24-OEM

Treiberprogramm für bis zu vier serielle Schnittstellen in IBM-kompatiblen MS-DOS-PCs und ATs, ideal zur Ein- und Ausgabe in Pascal, C, Basic oder Assembler (s. mc 10/87, S. 68):

- Resident laden mit CONFIG.SYS als DOS-Erweiterung
- Mit Dateinamen ansprechbar (RS1 bis RS4)
- Pascal: Read, Readln, write, writeln usw. verwendbar
- Basic: Wie gewohnt Open, Close, Print, Input usw.
- Baudrate 110 bis 38 400 Bd (!) einzeln einstellbar
- Auch bei 8 Bits noch Parity möglich (z. B. 8E1)
- Baudrate und Parameter auch nach OPEN noch änderbar
- Alle Steuerleitungen jederzeit setz- und abfragbar
- Status: Zahl d. Zchn. i. Puff., Parity-Fehler, Überlauf
- Je Schnittstelle bis zu 60 KByte Empfangspuffer
- Nur etwa 3 KByte Speicherbedarf je Schnittstelle
- Völlig transparent (durchlässig für alle Codes)
- Optional XON/XOFF-Protokoll

Mit Beispiel-Programmen in Turbo-Pascal, GW-Basic und 8086-Assembler und deutschem Handbuch . . . . . **198 DM**

software vertrieb

SHAMROCK SOFTWARE VERTRIEB GmbH

Telefon 0 89/59 54 68 + 69,
Karlstraße 35, 8000 München 2

SCHWEIZ: Microtron, Bahnhofstraße 2, CH-2542 Pieterlen, Tel. 0 32/87 24 29

ÖSTERREICH: PCS, Linzer Straße 210, A-1140 Wien, Tel. 02 22/94 41 79

NET.24-SERVER II

Mit NET.24 hat Shamrock Software die Vernetzung von PCs, ATs und Atari-STs für jedermann erschwinglich gemacht. Aber bisher war wie bei anderen Netzwerken der Server-Rechner „tot" – es konnte nicht gleichzeitig auf ihm gearbeitet werden. Das haben wir geändert! Das ist der neue Server:

- Für alle IBM-kompatiblen PCs, ATs, PS/2s
- Speicherresident ladbar und wieder de-installierbar
- Arbeitet vollkommen im Hintergrund
- Funktionen u. a.: Datei senden/empfangen/drucken, Directory anzeigen, MS-DOS-Befehle ausführen, File-Locking
- Externer Zugriff auf bestimmte Pfade einschränkbar
- Kompatibel mit allen Shamrock-NET.24-Produkten
- 4800 oder 9600 Bd Übertragungsgeschwindigkeit
- Wahlweise auf 5,25"- oder 3,5"-Disk, m. deutsch. Handbuch

Server II inkl. Adapter für serielle Schnittstelle . . . 198.– DM
NET.24-Teilnehmerkit (Adapter, Software, Handbuch) 98.– DM
5 m Verbindungskabel mit Cinch-Steckern . . . . . 7.98 DM
Starter-Kit: Server II, 1 Teilnehmer-Kit, 5 m Kabel **303.98 DM**

software vertrieb

SHAMROCK SOFTWARE VERTRIEB GmbH

Telefon 0 89/59 54 68 + 69,
Karlstraße 35, 8000 München 2

SCHWEIZ: Microtron, Bahnhofstraße 2, CH-2542 Pieterlen, Tel. 0 32/87 24 29

ÖSTERREICH: PCS, Linzer Straße 210, A-1140 Wien, Tel. 02 22/94 41 79

Ecosoft Economy Software AG

Postfach 1905, D-7890 Waldshut, Tel. 0 77 51-79 20
Villa Domingo, CH-6981 Astano, Tel. 0 91-73 28 13

Prüf-Software und Frei-Programme (fast) gratis

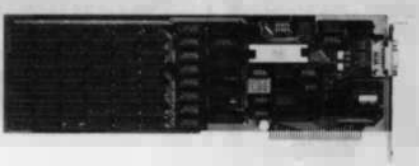
Über 3000 Disketten mit professioneller Prüf-Software (Programme vor Anwender-Registrierung prüfen), aber auch nützlichen Amateur-Programmen für den beruflichen und privaten Gebrauch für IBM-PC/Kompatible, Macintosh, Atari ST, Amiga, C64/128, Apple II.

Katalog auf Disketten und Verzeichnis DM 10.–

(Bitte Computermodell angeben und Banknote oder Scheck beilegen.)

Neu: Emulation von Fremdsoftware, z. B. MS-DOS auf Amiga, C64 auf Amiga, Macintosh auf Atari ST usw. Gratis-Info-Schrift verlangen. 389

Gegen Einsendung dieses Inserates erhalten Sie einen Gutschein für eine Gratis-Ecosoft-Diskette.



hochauflösende Farb-Grafik

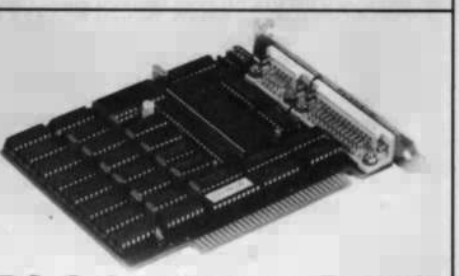
für IBM/XT/AT (u. Komp.) mit SC-64

- NEC-7220, 8 MHz Highspeed-Version
- 1024 × 1024 Pixel, 8 Farben, 3 Mbit RAM
- 64 MHz oder 40 MHz Videotakt
- weitgehend kompatibel zu Artist 1
- Softwaretreiber in Pascal, C, usw.
- SC-64 Grafik-Board **DM 2950.–** + MwSt.

Mehr als 500mal bei bekannten CAD-Systemen u. anderen grafischen Anwendungen im Einsatz., Made by SC.

SC SPECIAL COMPUTER GMBH

LW. Hauptstr. 5 D-7630 Lahr
Tel. 07821/4990 Telex 754848.



PC-Subprozessor-Board

für IBM (kompatible) PC, XT, AT

- CPU HD64180, Takt 4,6 MHz
- 512 KByte RAM, max. 4 KByte EPROM
- Zwei serielle Schnittstellen, vollduplex
- Input 5 Bit parallel (Maus, Lesestift usw.)
- Dipschalter (3 Bit, von Software lesbar)

Viele Anwendungen, z. B. intellig. Druckerpuffer, Barcode, Datenerfassung usw.

SC SPECIAL COMPUTER GMBH

LW. Hauptstraße 5 D-7630 Lahr
Telefon 0 78 21/49 90 Telex 7 54 848

OKI LASER- und MATRIXDRUCKER:
OKI LASERLINE 6 HP-Laserjet plus kompatibel . **3.695,–**
OKI 393 - 24 Nadeln, 450 Z/Sek. Epson LQ komp. . . **2.990,–**

PANASONIC MATRIXDRUCKER:
KX-P 1081 - 120 Z/Sek. IBM + ASCII Zeichensatz **495,–**
KX-P 1083 - 240 Z/Sek. neues Modell - sofort lieferbar **795,–**
KX-P 1592 - 180 Z/Sek. Breitformat 360 mm **1245,–**
KX-P 1595 - 240 Z/Sek. Breitformat 360 mm **1395,–**
Alle Modelle mit hervorragender NLQ Schrift 18x18 Matrix, Traktor und Walze für Einzelbl. serienmäßig!
Vollautomatischer Einzelblatteinzug für KX-P 1063 . . . . **390,–**
Auch die neuen **Panasonic 24-Nadeldrucker** mit Zubehör sind sofort lieferbar!. Bitte anfragen!

NEC Matrixdrucker:
Bitte fragen Sie nach unseren sehr günstigen Preisen - Wir liefern nur Originalgeräte mit Seriennummer und 1 Jahr Garantie

KABEL + DRUCKERINTERFACES
Druckerkabel 200 cm geschirmt an PC / AT **39,–**
Druckerkabel 200 cm beidseitig Stecker 36 polig. **49,–**
WW Interface 92000/G für C64 / C128 . . . . . **115,–**
WW seriell RS232/Centronics mit Buffer 8 KB **285,–**
Apple II Grafikinterface incl. Centronics Kabel **155,–**

MONITORE (Wir liefern alle Typen)
PANASONIC ST120 12" mit BAS Eingang **145,–**
PHILIPS BM 7513, grün, 25 MHz, TTL Eingang mit Kabel **289,–**
VICTOR 14" TTL mit Schwenkfuß, Flachschirm, mit Kabel sehr gute Qualität! Amber: **380,–** Grün: **360,–**

THEO WEBER ELEKTRONIK, 8700 WÜRZBURG
Eisenbahnstraße 22, Tel. 09 31-70 14 41

GETRONICS FM 1400, 14" TTL weiß mit Flachschirm **380,–**
GETRONICS MC 54 EGA Monitor der Spitzenklasse **1050,–**
NEC MULTISYNC - neues Modell **1495,–**

VICTOR COMPUTER
Wir führen das kpl. Programm von VICTOR zu unseren bekannt günstigen Bedingungen. Bitte fordern Sie unser Angebot an!
(VICTOR Produkte liefern wir nicht an Wiederverkäufer).

PANASONIC COMPUTER:
FX-600/A voll PC komp., mit 8086 CPU, Taktfreq. 4.77/7.16 MHz. Uhr/Kalender eingebaut. Floppy 360 KB, Centronics IF., große Tastatur mit sep. Cursorblock, MS-DOS und BASIC.
FX-600/A ohne Monitor und ohne Grafikkarte **1850,–**
FX-600/B mit 12" BAS Monitor und CGA Grafikkarte . **2095,–**
FX-600/C mit 12" TTL Monitor und Herk. kom. Grafik **2245,–**
Zweites Diskettenlaufwerk 360 KB - eingebaut - **295,–**

RAFI BTX DECODER
BTX PC/AT Einbaukarte mit Software und Modemkabel . **995,–**
BTX DEKODERTASTATUR - zum Betrieb mit FS-Gerät . . **975,–**

FESTPLATTEN / STREAMER - A-Qualität!
SEAGATE ST225 - 21 MB, 65 ms - meistverkauft! . . . **495,–**
SEAGATE ST238 - 33 MB, (RLL) 65 ms **575,–**
SEAGATE ST251 - 43 MB, 40 ms, mit Ontrac Softw. . **995,–**
SEAGATE ST4096 - 80 MB, 28 MB mit Ontrac **2250,–**
RODIME RO203E - 33 MB 65 ms, volle Bauhöhe **925,–**
OMTI 5520 PC Controller einschl. Kabelsatz **145,–**
WD 1002A-WX1 PC Contr. mit Superbios u. Kabels. . **165,–**
OMTI 8120 - AT Controller mit Kabels., Interleave 1! . . **345,–**

THEO WEBER ELEKTRONIK, 8700 WÜRZBURG
Eisenbahnstraße 22, Tel. 09 31-70 14 41

WD 1003 - AT Controller für zwei HD, mit Kabelsatz . . . **375,–**
GOLDCARD Einsteckplatte 21 MB, mit Lapine Platte . **845,–**
GOLDCARD Einsteckfestplatte 33 MB (RLL) **995,–**

ARCHIVE FASTAPE Backupsystem 20 MB für XT/AT **1395,–**
ARCHIVE FASTAPE Backupsystem 60 MB für XT/AT **1895,–**

PLOTTER + SCANNER
HANDY und MICROTEC Scanner . . . bitte anfragen!
SEKONIC SPL-410 A3 Plotter, 0,025 mm Schrittweite **2490,–**
(400 mm/Sek. voll HP/GL kompatibel)
SEKONIC SPL-430 A3 Friktionsplotter, HP/GL komp. **3290,–**
(mit Centronics und RS 232 Schnittstelle).
NC-Tablett ARISTO Digitalisiertablett A3 mit hoher Auflösung, einschl. Fadenkreuzcursor und Datenkabel **2490,–**

GRAFIKKARTEN - Sonstige Karten:
EGAWONDER - neue Universalkarte für alle Monitore **545,–**
> > jetzt auch mit HiRes Modis bis 800 x 560 Punkte < <
GENOA Super EGA HiRes - 800 x 600 Punkte - **645,–**
Paketpreis Genoa Super EGA HiRes mit GEM GRAF **795,–**
Addonics EGA/PGA/HGC max 640x480 . . . . . **495,–**
Grafikkarte **720x348** Herkules kompatibel **135,–**
Grafikkarte mit Centronicsport 720x348 - 12 MHz **245,–**
Uhrenkarte mit Kalender - mit Akkupufferung - **99,–**
Ein- Ausgabekarte 48 Bit parallel mit 3 Timer 8253 . **295,–**
AD/DA Wandlerkarte, 16 AD und 1 DA Kanal **398,–**

Bitte fordern Sie unseren umfangreichen Computer- und Zubehörkatalog sofort an! Wir liefern ein umfassendes Programm vieler führender Hersteller.

THEO WEBER ELEKTRONIK, 8700 WÜRZBURG
Eisenbahnstraße 22, Tel. 09 31-70 14 41

Der AT-Spezialist

OSIS-AT-Profi-System 12,5 MHz:
Mainboard mit 512 KB RAM bestückt (max. 1 MB RAM),
CPU 80286 (optional 80287), Taktrate 8/10/12,5 MHz,
Color-Grafik-Karte oder Monochrome-Grafik-/Printer-Karte,
FDD-Karte (Floppy-Disk-Controller),
1× NEC-Disk-Drive 1,2 MB/360 KB umschaltbar,
1× NEC-Disk-Drive 360 KB,
Tastatur AT 105 Tasten mit Cursor und Zehnerblock,
stabiles Stahlblechgehäuse, 200-W-Netzteil mit
rauscharmem Lüfter, ausführliches deutsches Handbuch
Preis mit Monitor ADI-DM-1400 bernstein **2925.–**
Preis ohne NEC 360 KB Disk-Drive
und ohne Monitor (ohne Abb.) **2199.–**

OSIS-AT-Profi-System 12,5 MHz:
Mainboard mit 512 KB RAM bestückt (max. 1 MB RAM),
CPU 80286 (optional 80287), Taktrate 8/10/12,5 MHz,
Color-Grafik-Karte oder Monochrome-Grafik-/Printer-Karte,
FDD/HDD-Karte (Floppy-Disk-/Hard-Disk-Controller),
1× NEC-Disk-Drive 1,2 MB/360 KB umschaltbar,
1× NEC-Disk-Drive 360 KB,
1× 20-MB-Hard-Disk Seagate ST 225 (65 ms mittl. Zgzeit),
Tastatur AT 105 Tasten mit Cursor und Zehnerblock,
stabiles Stahlblechgehäuse, 200-W-Netzteil mit
rauscharmem Lüfter, ausführliches deutsches Handbuch
Preis mit Monitor ADI-DM-1400 bernstein **3925.–**
Preis ohne NEC 360 KB Disk-Drive
und ohne Monitor (ohne Abb.) **3199.–**

Aufpreise:
Speichererweiterung von 512 KB auf 1 MB RAM ..... + 150.–
AT-Mainboard mit 512 KB RAM bestückt (max. 4 MB),
8/10/12,5 MHz umschaltbar, 0/1 Waitstate ..... + 139.–

Einzelpreise:
EGA-Monitor 14", High Resolution 1179.–
ADI-DM-1400-Monitor, bernstein
(für Monochrome-Grafik- und Color-Grafik-Karten) 427.–
ADI-Monitor, weiß
(für Monochrome-Grafik- und Color-Grafik-Karten) 493.–
NEC-1,2-MB-/360-KB-Disk-Drive 339.–
NEC-360-KB-Disk-Drive 299.–
FDD-Controller 149.–
FDD-/HDD-Controller 449.–
Color-Grafik-Karte 119.–
Monochrome-Grafik-Printer-Karte 139.–
ser./par. Karte 119.–
parallele Printer-Karte 58.–
EGA-Karte mit Hercules-Emulation 499.–
80386-System ..... Preis auf Anfrage
3½"-Disk-Drive ..... Preis auf Anfrage

Fordern Sie unser kostenloses AT-System-Info an!

RWL Computersysteme
Rüngeler & Wiggerich-Langhans GmbH

Höingstraße 7, 4750 Unna
Telefon (0 23 03) 17 58
Telex 8 227 764 osis d
Ihr Ansprechpartner:
Herr Schlüter

Technische Änderungen vorbehalten. Zwischenverkauf vorbehalten.
Lieferbedingungen auf Anfrage.

MARKT

Softwaregesteuerter Schnittstellenumschalter

Vier Peripheriegeräte kann mit dem „automatic line switch" ALS 10 ein Computer über die serielle Schnittstelle ansteuern. Die Umschaltbox wird über zwei mit Dip-Schaltern definierte aufeinanderfolgende Byte gesteuert. Die vom Computer kommenden Daten werden nicht direkt weitergeleitet sondern über ein 2stufiges Schieberegister geführt. Stimmen zwei Byte mit der eingestellten Sequenz überein, schaltet die Box auf den nächsten Ausgang um. Die Übertragungsrate reicht von 75 Baud bis 19 200 Baud.

IHP Electronic
Valterweg 10
6239 Eppstein 3
☎ 0 61 98/28 33

Auf vier Peripheriegeräte verteilt diese Box die Computer-Signale

Schach-Datenbank unter MS-DOS

Den Sprung vom Atari ST zu MS-DOS-Computern hat die Schach-Datenbank „ChessBase" geschafft. Pro Diskette sind maximal 5000 Partien unterzubringen, und zwar in einem komprimierten Code, der pro Halbzug nur 1 Byte beansprucht. Auch die automatische Klassifizierung von neu eingelesenen Partien nach Eröffnungen ist verfügbar. Chessbase ist kein Schachprogramm. Auf der anschaulichen Schachbrett-Darstellung werden keine neuen Partien gespielt, sondern nur die abgespeicherten nachvollzogen. Die Datenbank bietet jedem Schachfreund die Möglichkeit, sich in Ruhe alle wichtigen Partien, die mittlerweile lehrreiche „Schachgeschichte" sind, bequem zu studieren.

Frederic Friedel
Hauptstr. 28B
2114 Hollenstedt

Eine Stellung aus der Weltmeisterschafts-Partie Kasparov/Karpov (1986), aufgezeichnet in „ChessBase"

Software aus Rotterdam

Zwei interessante Programme für den Atari ST kommen aus Rotterdam. ST-Coder heißt das nur 25 DM teuere Programm, mit dem beliebige Dateien verschlüsselt werden. Das Besondere an dieser Software ist, daß der Schlüssel während der Verschlüsselung ständig geändert wird. In der entstehenden Datei können daher keine Regelmäßigkeiten entdeckt werden.
Das zweite Programm wendet sich an alle GFA-Basic-Programmierer, die direkt auf AES- und VDI-Routinen zugreifen wollen. Zu dem Programm, das 25 DM kostet, gehört eine deutsche Bedienungsanleitung. Oft benutzte Prozeduren werden in Bibliotheken abgelegt und vereinfachen so die Programmierung von größeren Software-Projekten. Auch eine Cross-Referenz-Liste und ein Ausdruck von GFA-Basic-Listings mit Zeilennummern werden von dem Programm GFA Proc-Linker angeboten.

HIQ-Systems
Postfach 21766
NL-3001 AT Rotterdam

Software-Betriebsstundenzähler

Eigentlich wie ein „Zeitschloß", das ab einem vorprogrammierten Tag den Zugriff auf ein Softwarepaket verhindert, arbeitet der Baustein DS 1207 von Dallas Semiconductor. Das Leasing beliebiger Software für eine bestimmte Zeit ist damit also ebenso möglich wie ein Probebetrieb für einige Tage. Auch das Kopieren der Software ändert nichts an der vorgegebenen Soll-Laufzeit, denn ohne diesen Chip, der wichtige Daten des Programms enthält, läuft nichts. Zuständig dafür ist ein 384 Bit umfassendes, batteriegepuffertes RAM, in dem diese gesicherten Daten untergebracht sind. Weitere 64 Bit enthalten einen Identifikationscode, um anzuzeigen, zu welchem Softwarepaket dieses Zeitschloß gehört, darüber hinaus arbeitet ein 64-Bit-Paßwort-Register. Beim richtigen Ansprechen des Bausteins erhält man also 384 Zusatzbits an korrekter Information, bei unautorisiertem Zugriff kommen lediglich Zufallsdaten heraus. Nach Ablauf der vorgegebenen Sollzeit wird der RAM-Inhalt selbständig gelöscht. Zu betreiben ist der Baustein am Drucker-Port eines PC/XT/AT.

Atlantik Elektronik
Fraunhoferstr. 11a
8033 Martinsried
☎ 0 89/8 57 00 00

Nach einer vorgegebenen Zeitspanne sperrt der DS 1207 den Zugriff auf ein zugeordnetes Softwarepaket

technik
dalvo

VERSION

10 MHz-0 WS

12 MHz-0 WS

Nach Landmark Vers. 2,04

-13 MHz

-15,6 MHz

Bis 1 MB (640/384), HDD/FDD Controller, 1 x RS 232, 1 x Centronics, superleises Gebläse, 1,2 MB Laufwerk, große Tastatur
Gehäuse: Baby I: 3 x slimline 5¼"
Baby II: je 2 x 3,5", 2 x 5¼"
Mini : 4 x slimline 5¼"
Baby II: in schwarz oder elfenbein
Netzschalter an der Front

Wir garantieren einen 30-stündigen Probelauf unter Einsatzbedingungen!

ANGEBOT DES MONATS

MONITORE 14" s/w flatscreen nur **DM 265,—**
12" amber nur **DM 225,—**
AT-TASTATUREN 105 Tasten, sep. Cursor, sep. Zahlenblock nur **DM 195,—**
XT/AT-FLOPPY-CONTROLLER
bis 1,2 MB (auch bei XT) **DM 122,—**

LAUFWERKE
1,44 MB YE-Data **3,5"** **DM 360,—**
720 KB YE-Data **DM 265,—**
1,2 MB YE-Data **5¼"** **DM 265,—**
360 KB YE-Data **DM 225,—**

EGA-SUPER-GENOA (Lizenzbau)
800 x 600 Auflösung **DM 495,—**
DT-PEGA „AUTOSWITCH"
640 x 480 Auflösung **DM 380,—**
GM-6 PLUS-MAUS
Mouse System/Microsoft Mouse
mit: Software „Dr. Halo"
Mouse Tasche, Mouse Pad **DM 159,—**

Wir bieten Ihnen interessante **Händlerkonditionen!**

DALVO TECHNIK VERTRIEB
Bärbel Dalheimer Erbacher Straße 37 6127 Breuberg 1
Telefon: 0 61 65/20 60 Telex 9 191 997 dtecd

```
02C0  E8 0D 00 E8 05 00 E8 0A 00 F9 C3 B4 09 CD 21 C3  ..............!.
02D0  E8 00 00 52 B4 02 B2 0D CD 21 B2 0A CD 21 5A C3  ...R.....!...!Z.
02E0  B4 0A BA 15 04 CD 21 BE 16 04 FC AC 3C 00 C3 52  ......!.....<..R
02F0  03 16 03 04 FA EC FB 5A C3 52 03 16 03 04 FA EE  .......Z.R......
0300  FB 5A C3 00 00 00 00 00 00 00 00 00 00 04 03 02  .Z..............
0310  01 07 06 09 05 09 00 00 00 00 00 00 00 00 00 00  ................
0320  00 53 75 6E 20 4D 6F 6E 20 54 75 65 20 57 65 64  .Sun Mon Tue Wed
```

So sieht ein Ausdruck des Programmpakets „Diffdump" aus: Eine präzise Lokalisierung unterschiedlicher Codesequenzen hilft, dem „Virus" auf die Spur zu kommen

Den Viren auf der Spur

Wenn sich vermeintlich gleiche Programme gar nicht gleich verhalten, dann liegen entweder – absichtlich oder unabsichtlich – manipulierte Versionen vor, oder es ist beim Kopieren bzw. Übertragen in einem Netzwerk etwas passiert, so daß Dateien anders ankommen, als sie abgeschickt wurden. Solche Schäden lassen sich mit etwas Mühe auch wieder beheben. Anders ist es bei „Virus-befallener" Software: Hier enthält das Programm „aufgepfropft" selbständig operierende Befehlssequenzen, die der Ursprungs-Software ein anderes (unter Umständen sogar ständig wechselndes) Verhalten aufzwingen. Und sollte gar das Betriebssystem infiziert worden sein, kann unabsehbarer Schaden entstehen. Als vorbeugende und bei dem geringsten Verdacht auf korrupten Datenbestand anzuwendende Maßnahme kann hier das Softwarepaket „Diffdump" dienen, das veränderte Dateien mit einer vorhandenen Original-Version vergleicht und auf recht komfortable Weise und schnell dem Anwender alle unterschiedlichen Bytes mit ihren Adressen aufzeigt. Besonderer Vorteil des Programms: die einfache Bedienung und die präzise Aufdekkung der Virus-Lokalisation. Der Preis für das Programmpaket beträgt 116 DM – eine Investition, die sich zum Schutz wertvoller Datenbestände oder Programme allemal lohnt.

Josef Seifert
Gerhard-Hauptmann-Ring 11/6
8000 München 83

Inside HyperCard

Unter dem Titel „Inside HyperCard I" ist die erste deutschsprachige Beschreibung aller HyperCard-Funktionen erschienen. Das Buch ist programmbegleitend zum Standardprogramm HyperCard für den Macintosh aufgebaut. Auf über 500 Seiten wird anschaulich jeder einzelne Befehl beschrieben. Die komplexen Funktionsabläufe werden durch die Konzeption des Buches leicht zugänglich und anwendbar. Immer wieder werden Bildschirmausdrucke im Text angegeben. Zur raschen Orientierung dienen ausführliche Register. Mit „Inside HyperCard II" bietet der gleiche Verlag einen Lehrgang zur Programmierung mit HyperCard an.

Format Verlag
Widdersdorfer Str. 236–240
5000 Köln 30
☎ 02 21/4 97 20 75

AD
Computertechnik GmbH

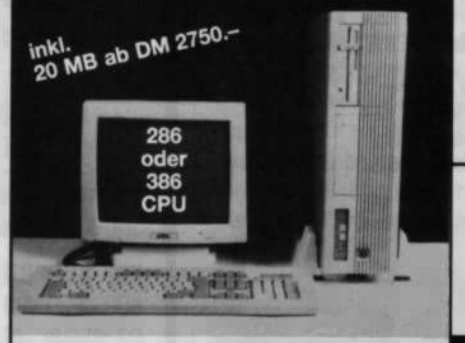
Halmstraße 3
2820 Bremen 77
Telefon 04 21/37 13 59
Fax: 04 21/6 36 52 96

Ladenverkauf:
Münchener Straße 58

11 verschiedene ATs vom AT-Profi z. B.:
10 MHz/0 **waitstates** ab **1799.-**
12 MHz, CPU-Card **999.-**
15 MHz, DSC ab **2299.-**
inkl. dynamischer Slot-Control
12 MHz/0 waitstates ab **2299.-**

3 × 80386
von
16 MHz ab DM **4999.-**
bis **26,9** MHz ab DM **6699.-**

Viele Einzelkomponenten preiswert, z. B. 10 MHz/0 waitstate, AT-Mainboard DM 699.–



AD Turbo AT

Alle AT inkl. Hercules-komp. Karte + Colorgrafikemulation, AT-Gehäuse für 4 Einschübe, Slimline-Schlüsselschalter, Resettaster, Turboschalter bzw. Waitstateschalter, parallelem Printerport, 200-W-Netzteil, RT-Tastatur mit 12 Funktionstasten – separatem Cursorblock (deutsch oder ASCII), NEC-Diskdrive 1,2 MB (liest und schreibt 360K), 80287-Sockel, 512K (max. 1 MB on Board).

Genius
GM-6 DM **89.-**
GM-6 plus DM 125.–
inkl. Dr. Halo, Menumaker
Maus inkl. Software

Handy Scanner **555.-**
Scanner + Texterkennung DM 1159.–
20 MByte **DM 479.–**

Turbo **XT** ab **929.-**

Einzelkomponenten, AddON, Netze, Drukker, Harddisk, AD/DA-Wandler, IEEE 488 und vieles mehr.

Sofort Katalog anfordern gegen DM 1.– in Briefmarken

Re-/Disassembler für Atari ST

Vollständig in Assembler programmiert wurde „Easy Rider", ein kombinierter Re- und Disassembler für den Atari ST. Programme, die mit Easy Rider reassembliert wurden, sind fast immer ohne Änderungen assemblierfähig. Selbst kompilierte Basic-Programme können korrekt reassembliert werden. Der erzeugte Quelltext hält sich an den Motorola-Standard. Das Programm wird mit einer knappen Anleitung für etwa 160 DM angeboten.

Belkenheid Computertechnik
Wellinger Weg 5a
4513 Belm
☎ 0 54 06/55 47

Portabel mit Kontrast

Den „Rasierspiegel"-Effekt mit der flauen Bildschirmdarstellung, wie sie von manchen Portabel-Computern her bekannt ist, will der Sharp-PC-7100 mit seiner „Super-twisted"-Flüssigkristall-Technologie vermeiden. Bei diesem Verfahren werden die Flüssigkristalle beim Übergang vom Hell- in den Dunkel-Zustand um rund 200° gedreht, was eine Kontrastverbesserung um den Faktor 2,5 erbringt. Deutlich verbessert wurde auch die Ablesbarkeit bei sehr flachem Betrachtungswinkel, wenn man also stark seitlich vom Computer sitzt. Und wenn es einmal recht dunkel im Raum sein sollte, hilft die zuschaltbare Hintergrund-Beleuchtung. Ansonsten kann dieser PC mit der 16-Bit-CPU 80C86-2, einem Arbeitsspeicher von 320 KByte (aufrüstbar), mit einem Floppy-Laufwerk sowie einer 20-MByte-Festplatte aufwarten. Je eine serielle sowie parallele Schnittstelle vervollkommnen diesen wirklich zur Leichtgewichts-Klasse zählenden Computer, der für knapp über 6000 DM erhältlich ist. Das lieferbare Zubehör-Programm umfaßt Speicher-Erweiterungen, Adapter für Zusatz-Monitore und einen ebenso tragbaren Thermodrucker.

Platzbedarf minimal – kontraststarke Flüssigkristall-Anzeige: PC-7100 von Sharp

Sharp Electronics Europe
Sonninstr. 3
2000 Hamburg 1
☎ 0 40/23 77 50

CeBIT '88 Hannover
Halle 7, Stand C 34 D 33

März Info

CHIP WISSEN

Günter Rolle:

Expertensysteme für Personalcomputer

Expert System Shells: Aufbau, Arbeitsweise, Marktübersicht

112 Seiten, Hardcover
30,– DM/ISBN 3-8023-**0194**-3

Der derzeitige technische Stand von PC erlaubt bereits Anwendungen der „Künstlichen Intelligenz". Wohl am bekanntesten sind die sog. Expertensysteme – Programme, denen man die Aufgabe menschlicher Fachleute zugedacht hat und die für IBM PC und Kompatible zur Verfügung stehen. Das Buch beschreibt, was Expertensysteme sind und wie sie realisiert werden. Ein Beispiel zeigt, wie man mit ihnen arbeitet und den Systemen das nötige Wissen vermittelt. Eine Übersicht marktgängiger Expert System Shells und ein Glossar der wichtigsten Fachbegriffe runden das Buch ab.

Hans-Peter Förster/Martin Zwernemann:

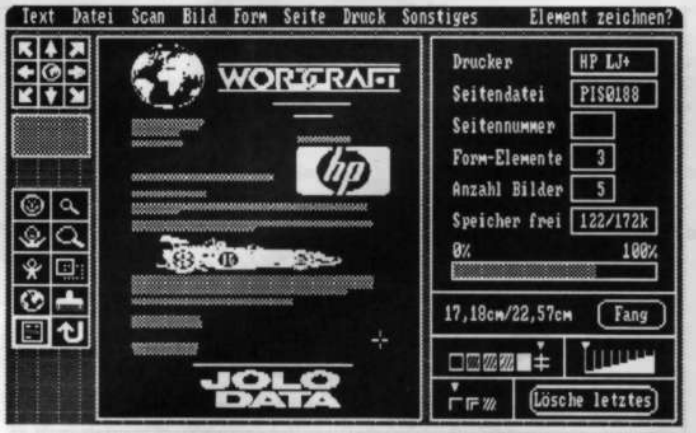
MultiMate 3.6 kurz und bündig

144 Seiten, 13 Bilder, Hardcover
33,– DM/ISBN 3-8023-**0208**-7

MultiMate ist ein komfortables, leicht bedienbares Textprogramm. Seine Kombination aus Menüsteuerung und Funktionstastenprinzip ermöglicht eine ganz auf das persönliche Vorgehen zugeschnittene Arbeitsweise. Das Buch beschreibt in kompakter Form die Funktionen und den Umgang mit MultiMate. Es richtet sich an interessierte Einsteiger, die schnell den Umgang mit diesem Programm beherrschen wollen. Alle MultiMate-Befehle stehen griffbereit zur Verfügung. Das Buch ist somit eine nützliche Ergänzung zu den Handbüchern.

Haben Sie schon den neuen „CHIP- Katalog 1988" ? Bestellen Sie gleich!

VOGEL Buchverlag Würzburg
Postfach 67 40 · 8700 Würzburg 1



In Wordcraft: Texte kann man mit Imagemaster-Grafiken einbinden

Wordcraft-Zusatz

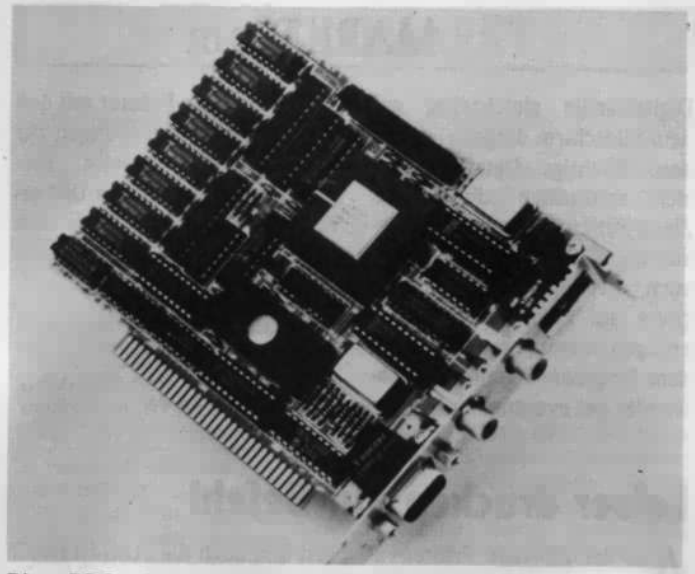
Zu einem eigenständigen Desktop-Publishing-Programm wird das Textverarbeitungsprogramm Wordcraft mit dem Zusatzprogramm Imagemaster. Beide Programme sind über Menüs in deutscher Sprache einfach zu bedienen. Imagemaster setzt eine Maus voraus und erlaubt das Einbinden von Grafiken und Fotos in Texte.

Mit der Formular-Funktion kann man eigene Formulare mit Linien oder Umrandungen in verschiedenen Strichstärken und Schattierungen gestalten oder mit graphischen Vorlagen kombinieren. Mit einer anderen Funktion werden Bilder oder Formulare quasi als Maske in den Laserdrucker geladen. Alle anschließend gedruckten Texte werden mit dieser Maske verbunden. Wer aus anderen Programmen Bildschirminformationen übernehmen möchte, kann eine Hardcopy erzeugen und mit Imagemaster verarbeiten. Imagemaster kostet für MS-DOS-Computer etwa 2000 DM.

Jolo Data
Bismarckplatz 18
3200 Hildesheim
☎ 0 51 21/5 56 20

Elf Grafikmodi auf einer Karte

Als ideale Ergänzung zu Multiscan-Monitoren bietet sich die Smart-EGA-Plus-Karte von Polyvision an. Elf Grafikmodi unterstützt die Karte für PC/XT/AT, darunter MDA, CGA, Extended EGA und VGA. Der Hercules-Standard wird in Farbe und monochrom unterstützt. Die Auswahl erfolgt auf DOS-Ebene über Befehle. An die Karte kann man vier verschiedene Monitore anschließen. Für wichtige Anwenderprogramme sind Grafiktreiber im Lieferumfang enthalten. Für Anwender von Spreadsheet-Programmen wie Lotus 1-2-3 oder Lotus Symphony bieten sich die Textmodi mit 132 × 25 Punkten und 132 × 43 Punkten an. In der Extended-EGA-Auflösung können sowohl 640 × 480 Punkte als auch 752 × 410 Punkte angesprochen werden. Die höchste Auflösung liegt bei 800 × 600 Punkten und ist für Programme wie AutoCAD, Ventura Publisher und PageMaker ideal. In allen Farbmodi kann man von



Diese PC-Steckkarte stellt elf Grafikmodi zur Auswahl

64 Farben 16 gleichzeitig darstellen.
Neben einem 9-poligen DB-Stecker ist die Karte noch mit einem RCA-typischen Stecker und einem Lichtstift-Eingang ausgestattet. Für die I/O-Portadreßauswahl ist ein Schalter vorhanden, der das BIOS-ROM abschalten kann. Die leistungsfähige Allround-Grafikkarte wird für 670 DM mit einer 3jährigen Garantie geliefert.

Polyvision
Georg-Brauchle-Ring 68
8000 München 50
☎ 0 89/1 41 50 77

PC-MOS/386 auf Demodisketten

Interessierte Anwender haben jetzt mit Demodisketten für das neue Multiuser-/Multitasking-Betriebssystem PC-MOS/386 die Möglichkeit, eigene Applikationen unter dieser Betriebssoftware zu testen. Der Rechner wird von der Diskette neu gebootet und lädt dann eine Demo-Sequenz in den Arbeitsspeicher. Anschließend hat der Anwender eine halbe Stunde Zeit, um PC-MOS/386 auf seine Bedürfnisse hin zu testen. Der Computer steigt nach 30 Minuten aus. Dieser Vorgang kann beliebig oft wiederholt werden, man bekommt aber dennoch die Sicherheit, daß die vorhandene Software problemlos läuft. Der Betrag von 35 DM für das Ansichtsexemplar wird bei einem eventuellen Kauf angerechnet.

Compucon Deutschland
Dachauer Str. 20
8066 Eschenried
☎ 0 81 31/7 00 10

Ihr mc-Jahrgang – sauber gebunden

Damit ihre zwölf mc-Hefte des Jahres 1987 auch weiterhin übersichtlich zur Hand sind, gibt es zum Preis von 12 DM eine mit Ganzleinen-Überzug versehene Einbanddecke. Bestellungen sind an folgende Adresse zu richten; auf Wunsch wird die Einbandarbeit gegen separate Berechnung übernommen.

Buchbinderei Hans Neumann
Lindenstr. 7
8151 Ober-Warngau
☎ 0 80 21/14 17

HANNOVER MESSE CeBIT'88
16. – 23. MÄRZ 1988
HALL 4,\FLOOR 1. OG,
STAND NO D08-E15.

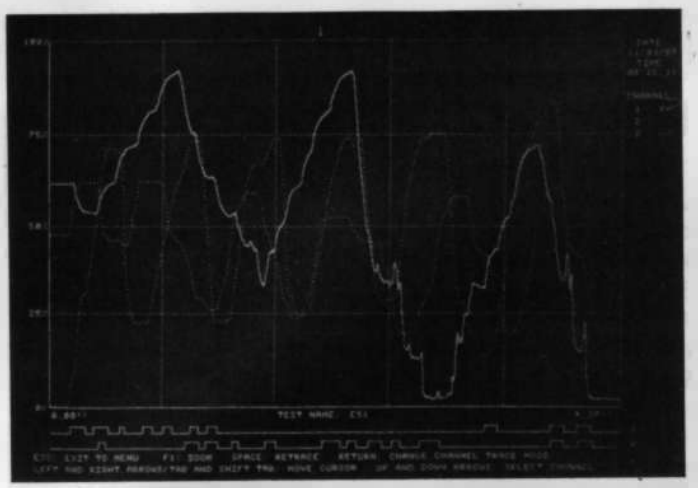
RABBIT 286 FCC APPROVED
DISTRIBUTOR WELCOME

HEAVY ON FUNCTION LIGHT IN WEIGHT
THIS RABBIT'S FOR YOU!

* NEW 640×400 DUAL MODE SUPERTWIST CGA/MGA LCD WITH EL BACKLIT, HERCULES PROGRAM WORKABLE
* "101+TURBO" MULTIFUNCTIONAL ENHANCED KEYBOARD. FOREIGN LANGUAGE VERSIONS AVAILABLE
* 1 MB ON BOARD, 3-1/2" 1.44 MB FDD AND 20 MB HDD WITH AUTOPARK
* EXTERNAL 5-1/4" DISKETTE DRIVE PORT
* ONE YEAR WARRANTY
* DIMENSION: 24×41×21 CM WEIGHT: 8.16 KGS

WITH ONE FREE BAG

Chicony
CHICONY ELECTRONICS CO., LTD.
7FI., NO. 35, KUANG FU S. RD., TAIPEI 10552, TAIWAN, R.O.C.
TEL.: 886-2-764-7277 (REP)
FAX: 886-2-7617237, TLX: 14465 CHICONY



Drei Analog- und zwei Digital-Kanäle (ganz unten) wurden hier mit dem Programmpaket „Pacan" aufgezeichnet und dargestellt

Übersichtliche Meßdiagramme

Sowohl CGA- als auch Hercules-Monitore bedient das menügesteuerte Meßdatenerfassungs-Paket „Pacan". Der Anwender kann damit nach vorheriger Parametereingabe sowohl Analog- wie auch Digital-Meßdaten von maximal zehn PC-Erweiterungskarten erfassen und verarbeiten. Die Aufzeichnung von bestimmten Signalereignissen wird dabei durch umfassende Datenwort- oder Analogpegel-Triggerung erreicht, nach der Datenspeicherung können gleichzeitig zehn analoge Kurvenzüge (wie beim Oszilloskop) sowie zwei Digitalkanäle gleichzeitig auf dem Bildschirm dargestellt werden. Wichtige Details lassen sich schließlich durch die Zoom-Funktion lupenartig herausvergrößern. Alle Eingaben, auch diejenigen zum Abspeichern auf Floppy oder Platte, erfolgen menügesteuert. In jedem Eingabefeld steht dem Anwender bei eventuellen Unklarheiten ein Help-Fenster mit den notwendigen Erläuterungen zur Seite. Das komplette Programmpaket ist für 789 DM erhältlich, eine Demoversion kann für 25 DM bestellt werden.

appli-data
Hauptstr. 136
7582 Bühlertal
☎ 0 72 23/7 45 46

Leiser drucken auf Befehl

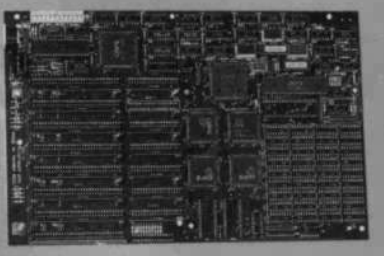
Aus der 24-Nadel-Printer-Familie bringt Star jetzt den LC24-10 heraus. Er verfügt über vier eingebaute Schönschriftfonts, schafft 170 Zeichen/s im Normal-Modus bzw. 57 Buchstaben/s mit Letter Quality. Erwähnenswertes „Beiwerk" sind ein automatischer Papier-Vorschub/Rückzug mit Trenn-Automatik zum einfacheren Trennen von Endlosformularen ohne Papierverlust und auch die „Leise-Taste", mit der die Anschlag-Frequenz der Nadeln halbiert wird, um beispielsweise beim Telefonieren den Geräuschpegel zu senken – allerdings bei gleichermaßen halbierter Druckgeschwindigkeit.

Star Micronics Deutschland
Mergenthalerallee 1–3
6236 Eschborn
☎ 0 61 96/7 01 80

BEST DESIGN & MANUFACTURE

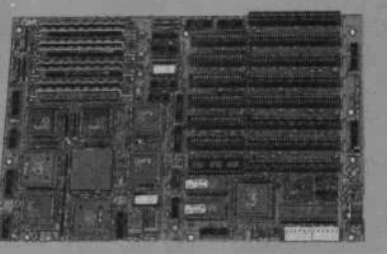
ECS-286

* 6/8/10/12 MHz
* 80286-10 CPU
* 640/384K RAMDISK
* 4 layers P.C.B.
* 1MB RAM expand on BOARD
* legal AWARD BIOS



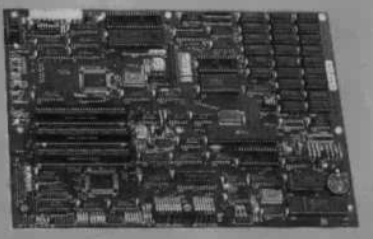
ECS-386

* 16MHz 80386-16CPU O Wait State (20MHz OPTION)
* Up to 8MB RAM on BOARD
* BABY AT size
* Fully AT compatible
* Optional 80287 CO-PROCESSOR



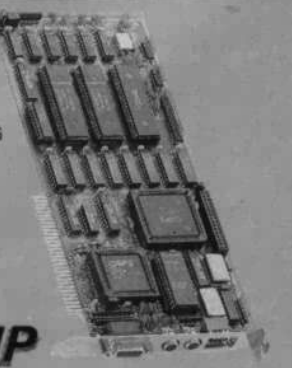
ECS-M88 (ALL IN ONE 8088 MOTHER BOARD)

* 8088-1 CPU mother BOARD (4.77/10M MHz)
* ONE PARALLEL (PRINTER PORT)
* SERIAL PORT (RS-232)
* GAME PORT
* LIGHT PEN
* MICROSOFT BUS MOUSE INTERFACE
* COLOR DISPLAY COMPATIBLE (ADDITION FEATURE: 640* 400 DOUBLE SCA DISPLAY MODS)
* HERCULES GRAPHICS COMPATIBLE
* 2x360K FDC connector
* 1x1.2 MB FDC connector



ECS-V100

* 100% IBM EGA CARD COMPATIBLE
* COMPATIBLE WITH: IBM COLOR GRAPHICS™, IBM MONOCHROME™, HERCULES GRAPHICS™, IBM PROFESSIONAL GRAPHICS™,
* 256K BYTES DISPLAY MEMORY
* HARDWARE SUPPORT FOR CONTEXT SWITCHING
* ADJUST AUTOMATICALLY TO DISPLAY MODE
* TWO TIMES FASTER PERFORMANCE OVER IBM EGA WITHOUT SOFTWARE CHANGES
* ONE PARALLEL PORT (PRINTER PORT)
* TWO SERIES PORT (RS-232)



OEM WELCOME
OEM WELCOME
OEM WELCOME
OEM WELCOME

ELITEGROUP
Computer Systems Co., Ltd.
5F-1, No. 77, Sec.2, Chi Lung Rd., Taipei, Taiwan, R.O.C.
Tel:(02)7027684, 7542477-8 FAX:(02)7359080 Telex:16499 ELITEGO

mc-MINIMARKT

CAD-Computer, AT286, 1 MByte, Gehäuse IBM. High Resolution, Videokarte, 1×Floppy, 72-MByte-Platte (C, D, E), seriell/parallel, Mouse, Tastatur, Colour-Bildschirm, System hat noch 3 Mon. Garantie, DM 5500.–.
Chiffre: mc 096

IBM-Farbmonitor, VB 800.– DM. ☎ 0 70 71/2 47 98

CBM-8000-Soft- u. -Hardware, Restposten. Thomas Klaus, Weinberg, 7164 Mittelfischach

2. Hand mit Garantie
CBM 8296-D – Anfrage
CBM 8032, 8296 – Anfrage
CBM 8050, 8250 – Anfrage
CBM 3040/4040 – Anfrage
CBM 4022/3022 – Anfrage
CBM-Interfaces, CBM-Kabel
Bondwell 12/14 portable
kpl. mit sehr viel Software
Commodore Reparaturdienst
Mietkauf/Leasing/Ratenkauf
Branchenlösungen für PCs
BÜCOTECH,
Hauptstraße 167m,
5020 Frechen,
☎ 0 22 34/1 56 92 G

Fabrikneuer Mac+, Imagewriter II, ext. LW, viel SW (Text, Graphik, Datenbank), sehr günstig. Hoffmann, ☎ 08 21/48 21 33

LOGIK-ANALYSATOR, Hewlett-Packard hp1610A, 32-Kanal, DM 2850.– VB. ☎ 0 62 98/32 28

Gewerbl. Anzeigen sind mit G gekennzeichnet.

NEUERÖFFNUNG
COMPUTER-FACHMARKT
IN HAGEN
Unser Angebot zur Eröffnung:
Turbo XT, 256-KByte-RAM, 4,77/8 MHz, CGA oder HGC, 360-KByte-LW, 150-W-Netzteil, XT-Tastatur, Klappgehäuse, 912.– DM.
Turbo AT, 1-MByte-RAM, 6/10 MHz, 180-W-Netzteil, HGC, 1,2-MByte-LW, AT-02-Tastatur, Klappgehäuse, 2275.– DM.
Die Garantie beträgt 12 Monate.
COMPUTER FACHMARKT,
Inh. A. Bröker,
Enneperstr. 16, 5800 Hagen,
☎ 0 23 31/40 71 47 G

NDR-Computer, CPU 68 KByte, GDP 64 KByte, 3×R OA64 mit Mon. 68 KByte u. 16-KByte-RAM, CAS, Prommer, IOE, 400 DM.
☎ 0 51 45/65 94

MIP-MAUS-Programm laden und jede Software arbeitet mit Mausteuerung (z. B. dBase, Lotus), Preis 99 DM. Händleranfr. erw. Kayser, ☎ 0 52 58/30 63

SIEMENS-PC 16-11 mit 384-KByte-RAM, 1×FD, 10-MByte-HD, viel SW, sehr wenig b. kompl. 1600 DM. ☎ 0 89/6 97 30 73

MC-CP/M-System, prof. aufgebaut, SYS1, OUT1, FLO1-D, TERM1, Schruff-Geh. mit Schaltnetzteil, Profi-Tastatur ACS-AN-95FI, 2×5¼"-BASF-Drucker, Unterl., viel Software, VB 1600.– DM.
☎ 0 91 70/85 16, ab 17 Uhr

EPROMSIMULATOR f. 2716 bis 27512 (o. Gehäuse), **248.– DM.** Info bei: **HüWiCo,** Am Siepen 17, 4630 Bochum, ☎ 02 34/36 12 06 G

PHILIPS P2010, CP/M, 64-KByte-RAM, 32-KByte-Videospeicher, 2×160-KByte-Disklaufw., 2× ser. S., Monitor, Tastatur, WordStar, Calcastar, Basic, Turbo-Pascal, 900.– DM. ☎ 0 52 51/6 24 33

Zwei 8"-Disketten-Laufw., Techmar-Controller f. IBM-PC, Software f. alle Formate, 4"-Disketten, 800.– DM.
K. Herrig, ☎ 0 61 28/68 64

Preiswerte Erweiterungen für: Gepard, kws, VME, c't 68, HSK, ATARI ST, AMIGA, KIT 2 u. a.
– 1-MByte-Dyn.-RAM-Karten
– 2-MByte-EPROM-Karten
– 1-MByte-Statische-RAM-Karten
– Farbgrafik-Karten
– A/D-Wandler-Karten
– D/A-Wandler-Karten
– Relais-Karten
– I/O-Karten
– 680XX-Steuercomputer
– DIN-A3-Farbplotter
– Festplatten
– Floppystationen u. Laufwerke
– Drucker und Monitore
Kostenlose Information bei:
HSK-Elektronik, H. S. Kiefer
Castroper Str. 129 + 148,
4600 Dortmund 15
☎ 02 31/33 40 91/33 36 67 G

SIEMENS-PC-X, Floppy 650 KByte, Harddisk 13 MByte, Arbeitssp. 1 MByte, Liz. SW. **SINIX,** Multiplay, Hit, Informix, **6000 DM.**
Chiffre mc 94

MIDI für den Apple II! Neues wesentlich erweitertes Interf. und 4-Spur-Sequencer. Info geg. 1.30 Porto, J. Lange, Maßmannstr. 4/3, 8000 München 2

mc-BASIC-EMUF, LCD-DISP., 32-KByte-RAM-Netzteil, betriebsbereit, 350.– DM.
G. Müller, ☎ 02 28/25 29 17

NDR-Klein-68K: 18-Karten-NT, Dru., Mo., Flodr. CP/M Bas. Pas. Gos. usw. Lit: ~50 % BSP.
☎ 07 11/44 54 22, ab 19 Uhr

NDR-Computer, CP/M 2.2/3.0, kompl. System in Schroffgeh. 2×1 MByte, 3.5", Bernst.-Mon., Tast., viel Softw., NP ü. 3000 DM, VB nur 1100 DM.
☎ 0 22 73/5 17 80, abends

MC68000, komplett mit Monitor, PC-Tastatur + Gehäuse, zusätzl. par./ser. Schnittstelle, 1 LW 720 KByte + 1 LW 1,2 MByte, VB 2000.– DM. ☎ 02 21/5 90 64 57

DATENVERLUST?
PROGRAMM-ABSTURZ?

SDF 6E
220 V/6 A
einstufig

DM 99,75

Steckdosenfilter und Netzfilterleisten von MWB schützen vor netzbedingtem Datenverlust oder Programmabsturz.

FL-2-6D
220 V/6 A
zweistufig

DM 193,80

MWB MESSWANDLER-BAU AG
Nürnberger Str. 199
8600 Bamberg
Tel. 09 51/18 03-2 17

sind die Folgen für Ihre EDV unangenehm und teuer – manchmal sogar katastrophal.

Schützen Sie sich wirksam. Mit DVS-Notstromgeräten. Immer zuverlässig und sicher. Und viel preiswerter als Sie denken.

DVS
Datentechnik GmbH
Ludwig-Thoma-Straße 1b
8034 München-Germering
Telex 524966
☎ 089/8419064

SKC DISKETTEN
100 % Fehlerfrei
5¼
3½
zoll

Unbedingt
Information und Angebot anfordern

SCHUKAT
Für Industrie und Fachhandel:
Krischerstraße 27 · 4019 Monheim / Rhld.
Tel.: 0 21 73/5 00 05 · Telex 8 515 732 sele d

Macintosh

Apple II

MS-DOS

HARDWARE SOFTWARE ZUBEHÖR BÜCHER
Spezialkatalog für ☐ Apple II, ☐ Macintosh, ☐ MS-DOS
Fordern Sie unter Angabe Ihres Rechnertyps den entsprechenden **Gratiskatalog** an!
pandasoft Dr.-Ing. Eden
Uhlandstr. 195 D-1000 Berlin 12
Tel.: 030 / 31 04 24
Telex: 185 859
BITTE SCHICKEN SIE MIR IHREN GRATISKATALOG ZU!
Name: ..... Adr.: ..... Rech.Typ: ..... MC

mc-MINIMARKT

BIETE AN HARDWARE

Prozessorbaustein neu, Intel 80-287-3, bis 8 MHz, Stück 200.–.
☎ 0 80 67/5 71

NDR-Comp., Einsteigerpaket: POW5V/SBC2/HEXIO/BUS2/IOE/ Christiani Grundkurs, VB 300.–.
☎ 09 11/44 90 31

Apple 2+, 80Z, IBS-AP22/34, ser./par., Pol., div. andere Karten, 2 LW 40 Track, 1 LW 2×80 T, Monitor, Unterlagen, VP 1500 DM, Fx 80+ 750 DM. ☎ 02 34/49 67 18

Apple II: RAM-Disk 256-KByte, Z80B, 2 Disk 640 KByte u. Ehring-controller. Suche ser. Card.
☎ 0 70 71/6 82 64

MC68000, 2,2-MByte-RAM, **Z80-Karte,** (4/8 MHz), 2×FDD, 200-W-NT, AT-Tastat., Maus, SASI-Contr., Uhr, SW, VB 3900.– DM.
☎ 09 06/17 81

SHARP MZ-731, Z-80A-Rechner, 64 KByte, Kassettenrecorder, 4-Farben-Plotter, Uhr, Musikfunktion, Basic und Spiele, VB 400.– DM. ☎ 0 52 51/6 24 33

CBM 8296, 8250/p, 4022, Software, kompl. 2000.– DM.
☎ 0 79 73/51 26

Magnetkartenleser
☎ 05 71/4 98 56 Ⓖ

IBM-Kompatible komplett: DER AT **80386 komplett 8700.–,** Mini AT 2100.–, XT 1600.–, 30 MByte 740.–, NEC-Multisync 1380.–, Händleranfragen erwünscht.
☎ 0 63 03/43 87 Ⓖ

286-Speed-Karte nur 798.– DM, 10 MHz und 8-KByte-Cache-Speicher.
Ihr XT + Comp. schneller als AT! Hdl.-Anfragen erwünscht!
FSM-Trogisch, 3051 Suthfeld, Gartenstr. 10, ☎ 0 57 23/8 19 46 Ⓖ

C't-Text-Terminal, 5 Zoll, 2×80 Drive, MS-DOS/CP/M-Rechner 80186 (Slicer), IBM-kompatibel, kompl. oder Einzelteile abzugeben. ☎ 0 52 32/6 10 10

1 Typenraddrucker, TRD170S, neu, 2 Farbbänder, 560.– DM.
☎ 0 22 93/21 88 Ⓖ

MC68000, 1.5-MByte-RAM, Rhothron-Programmer, IBM-Kbd, 2 Shugart 5.25", RAB, XED, FBASIC (neu), div. Preis VB.
☎ 0 53 03/49 00

MC68000, 512-KByte-RAM, 1 LW, komplett mit Geh., Tast., betriebsbereit, 600.– DM.
☎ 0 66 41/6 15 39

NEU IN ANDERNACH: PC-/XT-/AT-Kompatible zum Anfassen und Testen und vieles mehr. Computer-Corner. ☎ 0 26 32/4 31 19

Taylorix 59/MPE incl. Programm FIB 64, LOG 94, PARF, kompl. m. Drucker, Tast., Term., Zubeh., geg. Höchstgebot zu verkaufen.
☎ 0 89/53 27 56

HARDWARE-MESSWERT-ERFASSUNG für fast alle Computer (siehe mc-Quickie).
L. Bockstaller Ⓖ

Gewerbl. Anzeigen sind mit Ⓖ gekennzeichnet.

SCHRITTMOTORSTEUERUNG
XYZ-Achsensteuerg. f. Computer mit Parallelport. Mit Netzteil und 3 Schrittmotoren **DM 269**
BOHRPROGR./C64 DM 98
Schrittmotor DM 29. Info DM 2.
PME, Hommerich 20, 5216 Rheidt, ☎ 0 22 08/28 18 Ⓖ

SEIKOSHA MP-1300AI Drucker, 300 Z/sec, 10-KByte-Puffer, serielle u. parallele Schnittst., auch IBM-Zeichensätze, NLQ, Traktor, Einzelblatteinzug, VB 1000.– DM.
☎ 0 52 51/6 24 33

Verkaufe orig. APPLE IIe, 2 Laufwerke, 80Z + 64 KByte, grüner Monitor, Joystick, par. Schnittst., viel Literatur, Software, 150 Disketten + 3 Disk-Boxen, VB 1800.–.
☎ 0 89/60 43 74

20-MByte-Festplatte incl. Contr. sowie EPSON-Drucker und IBM-XT-komp. Computer.
☎ 09 21/4 45 20, ab 18 Uhr

TAUSCH

Tausche **VME-Bus-System,** OS9/68K V2.0, 1-MByte-RAM, 20-MByte-HD incl. EPR.-Prog. gegen IBM-AT-komp. mit HD oder verkaufe auch f. DM 10K.
Wer braucht noch **CP/M+-System,** 2×8"-Floppy slim-line, 384-KByte-RAM, Grafik, ECB-System mit Prommer? Alles in Schroff-compaq-Geh., **billigst** abzugeben. Raab, ☎ 0 70 21/57 31 85, tagsüber

KONTAKTE

Erstelle preisw. Beschr., Handbücher, Datenblätter und Dokumentationen in Offsetdruckqualität. Chiffre mc 095

Preise wie im Paradies!

EPSON
LQ-500 798,-
LX-800 538,-
LX-800 VC/P 638,-
FX-800 919,-
LQ-850 1248,-
LQ-1050 1648,-

NEC
P 2200 848,-
P6 999,-
P7 1248,-
P6 Color 1298,-
P7 Color 1548,-
P9 XL 2999,-

Commodore
Amiga 2000
m.Mon.1084 2798,-
PC-10 III 1699,-
PC-10 III 2/30 2348,-
PCAT 40/40 5098,-
PCAT 60/40 9598,-
PCAT 60/144 10999,-

Seagate
20 MB Kit 548,-
30 MB Kit 588,-
40 MB (40ms) 768,-
40 MB (28ms) 898,-
20 MB ST-125-0
3,5' (40ms) 528,-
ST-125-1(28ms) 618,-
50 MB ST-157R-1
3,5' (28ms) 899,-
80 MB (28ms) 1549,-

Star
Neu! Neu!
LC-10 498,-

Seikosha
SL-80 AI 748,-
SL-80 VC 768,-

Olivetti
M240/55G 3338,-
M240/0520G 3938,-
M28 (20MB) 4598,-

Citizen
C 120 D 398,-

Okidata
ML-393 2048,-
ML-292 949,-

Plantron
PT-XT 1678,-
PT-AT 2498,-
Knüllerpreise:
Baby AT 3448,-
PT-386/80 6866,-

Schneider
LQ-3500 768,-
PC-1640
ECD/HD20 3649,-

Zubehör
Mitsubishi Multisync
EGA Monitor 1298,-
EGA Wonder 398,-
Super Genoa 418,-
Alloy Streamer 748,-

Computer Discount 2000 GmbH
Hinter der Bahn- 5403 Urmitz-Bhf.- Tel.02630/84227 - TTX 2630915

disys
IBM PC, XT, AT, PS/2 Instrumentation
disys Meß- und Testsysteme GmbH
Auf der Grefenfurth 1-3
D-5064 Rosrath 5
Telefon: 0 22 05 / 8 40 19
Telex: 887 3768 disy d

Multifunktionskarte 16-64 Kanal A/D, 12/16 Bit, D/A 12 Bit, 20 Bit I/O, 2 Relais

HARDWARE:
16/12 Bit Auflösung
1,5-50 µs Wandelzeit
1/10/100/1000-fache Verst.
25-70 KHz Abtastrate bzw.
16.000 bis 50.000 Kanäle/sec.
PT 100, Thermoelemente, Brücken
Meßbereich von µV bis max. 10V DC

PCI-00

SOFTWARE:
Trerber-Software für z.Zt. 11 Hochsprachen
Datenlogger-Software-Paket tabellarische und grafische Darstellung der Meßwerte mit Statistik, wie X, S, Min, Max, etc. einfache, praxisnahe Bedienung.

Eprommer Eprommer
mit ext. Textoolsockel im Gehäuse., Editor, Programmierspannung:: Vpp 25V; 21V und 12.5V Programmiert 2716 bis 27512 auch C Typen
Lieferumfang: Steckkarte ext. Epromsockel, Software Beschreibung
zum Preis von nur **248 DM**

Frei Programme
für IBM oder Kompatible Computer in deutsch und englisch weit über 1200 Disketten.
Liste der Freiprogramme KOSTENLOS.

TEAC 55BR 360 KB **270DM** **35FN-23** 720KB 5,25"Rahmen **339DM**
TEAC 55GFR 1,2MB **289DM** **35HFN** 1,44MB **279DM** 3/88
TEAC 55FR 720 KB **289DM** **35FN** 720KB **269DM** Hanbuchje 15DM
Handy Scanner 698DM **HandyReader** Texterkennung **998DM**

UEDING electronics
Holtewiese 2 Tel. 02373 63159
5750 Menden 1 DFÜ 02373 66877 8n1
Inh. Gregor Ueding Gesamtpreisliste mit weiteren preisgünstigen Angeboten anfordern

EPROM-LÖSCHGERÄTE · NEU: KOMPL. BAUSÄTZE

L 6

NT 6

B 12

Alle Fertiggeräte mit Sicherheitsschalter, Netzbetrieb, Löschzeit 5 min, Röhre u. Starter auswechselbar.
Neu: mit aufgedruckter EPROM-Tabelle.

Für 6 EPROMs:
N 6 DM 118,– (Timer nachrüstbar)
NT 6 DM 148,– (mit Timer)
Für 12 EPROMs:
N 12 DM 138,– (Timer nachrüstbar)
NT 12 DM 198,– (mit Timer)
Für 24 EPROMs:
NT 24 DM 248,– (mit Timer)
Für 36 EPROMs:
NT 36 DM 298,– (mit Timer)

Bausätze:
FT 6 Kompl. Löschgerätebausatz mit Gehäuse, Timer (bis 15 min), Sicherheitsschalter, Netzbetrieb, inkl. aller Montageteile.
für 6 EPROMs DM 89,–
F 6 (ohne Timer) DM 69,–
FT 12 DM 119,–
wie FT 6, jedoch für 12 EPROMs
F 12 (ohne Timer) DM 99,–
L 6 DM 49,–
Löschset ohne Gehäuse (s. Abb.) nur Bauteile (s. Abb.):
B 6 DM 39,–
B 12 DM 49,–
TM 2 (Timer einzeln) DM 49,–

HEINZ WELTER
GERÄTETECHNIK
Borkenwirther Str. 40
Postfach 3029
4280 Borken-Weseke
Telefon 02862/1505
Postgiro Dortmund
254883-463

mc-MINIMARKT

Wir suchen für sofort aufgeschlossenen und dynamischen Fernsehmeister mit fundierten Kenntnissen in Colour- und Videotechnik.
☎ 0 74 53/89 75 [G]

HX-20-CLUB und PC-PROFI-GRUPPE c/o Jürgen und Susanne Miersch, Hindenburgd. 58a, 1000 Berlin 45

VERSCHIEDENES

Suche für AT-Wartung die vom Gerät angegebenen Fehlernummern mit Fehleranalyse.
Günter Opatz,
Bülowstr. 3, 3000 Hannover 1

Suche Sonderheft mc „MODEM" Nr. 87 oder Fotokopien.
☎ 07 11/35 29 52

Eigener Computer bringt Geld, 24seitige interessante Info, 20 DM + 1.40 Porto. G. Hein, In der Worth 12, 3155 Edemissen

Transfer IBM EBCDIC Mainframe – PC-1/2"-Tape in ASCII oder dBASE, DM 60.– pro MByte.
☎ 0 89/9 03 44 64

Entwicklung und Fertigung elektr. Baugruppen, Layouterstellung für Platinen. ☎ 0 51 46/86 81 [G]

CAD-LAYOUTENTWICKLUNG Modernste Technik – verschiedene Programme – dadurch optimale Kundenanpassung – SMD-Filmherstellung. KEMPF-Layoutentwicklung, Willy-Andreas-Allee 3/Zi. 615, 7500 Karlsruhe

Biete mc 3/85 bis 11/87 und c't Jg. 87. ☎ 0 30/4 95 56 89

JOYCE → PC, JOYCE → ATARI CP/M → PC, CP/M → ATARI Info: Bernd Drost, Schulstr. 67, 6382 Friedrichsdorf,
☎ 0 61 75/6 04 [G]

8. Nürnberger Flohmarkt für Amateurfunk- und Computerfans, 23. April 1988 v. 8.30–17.00 Uhr im Nürnberger Messezentrum. Private u. profess. Anbieter. Über 5000 qm Ausstellungsfläche. Einheitspreis: DM 20.– p. Tisch. Ausstellerinfo: Hans Kammler, Laurentiusstr. 9, 8500 Nürnberg 60,
☎ 09 11/64 44 34 [G]

Leiterplatten: Sofortdienst für Muster, Layouts, Entwicklungen.
☎ 0 77 02/22 88 u. 18 84 [G]

Die 2. Auflage des ZX-Hardwarebuches ist da! **Starke Schrittmotore!** Schrittmotor-Steuerung Katalog 6/87 gegen DM 5.– in Briefmarken. Decker & Computer, Postf. 967, 7000 Stuttgart 1 [G]

STAUBSCHUTZHAUBEN – EIN PROBLEM WENIGER – z. B. für IBM-PC und Kompatible, zweiteilig (Zentraleinheit mit Monitor und Tastatur).
NUR DM 40.– zzgl. Versandk. Gesamtpreisl. DM 3.– Briefm.
K & S-VERSAND
K. Schellhammer, Dorfstr. 26, 2061 Kleinwesenberg
☎ **0 45 33/35 66** [G]

Durchkontaktierte Leiterplatten; Oberfläche Pb/Sn ungeschmolzen, Klein- und Mittelserien, Muster in wenigen Tagen, auch einseitig. **Bestückungen – Layouts.** M. Ernst, Elektron. Systeme, Pf. 8 41, 5800 Hagen 1,
☎ 0 23 31/40 30 07 [G]

mc bis 12/86, DM 70.–.
☎ 0 81 51/1 31 07

Völlig neu überarbeitete Ausgabe, über 1280 Seiten stark! Mit ca. 3000 Abb. und ca. 600 Schaltungen, Plänen, Skizzen, Anschl.-Bildern. Schutzgeb. 16,- DM. Versand: Vorkasse Inland 19,- DM (inkl. Porto). Postgirokonto München, 244822-802. Nachnahme Inland 22,20 DM (inkl. NN-Gebühr).
RADIO-RIM GmbH, Bayerstraße 25, 8000 München 2,
Póstfach 202026, Telefon (089) 551720. Telex 529166 rarim d

Auftrag für Gelegenheitsanzeige in mc

Anschrift: Franzis-Verlag GmbH, Anzeigenabteilung, Karlstraße 41, 8000 München 2

Name und Adresse _____

_____ _____
Unterschrift/Ich zahle sofort nach Rechnungserhalt. Datum

☐ Preis für private Gelegenheitsanzeige je Druckzeile 4.– DM. Chiffregebühr je Anzeige 8.– DM.

☐ Preis für gewerbliche Gelegenheitsanzeige je Druckzeile 14.82 DM (dürfen nicht unter Chiffre erscheinen). Bitte beachten Sie, daß **Preise in Anzeigen die MwSt.** enthalten müssen.

mc-programmbörse
☐ suche Software
☐ biete an Software

mc-minimarkt
☐ suche Hardware
☐ biete an Hardware

☐ Tausch ☐ Verschiedenes
☐ Kontakte ☐ Chiffre-Nummer
☐ vollständige Adresse
☐ Telefon

Ich wünsche folgenden Text in der nächstmöglichen Ausgabe zu veröffentlichen:

(Zutreffendes bitte ankreuzen)

Bitte mit Schreibmaschine oder in Druckschrift in Groß- und Kleinbuchstaben ausfüllen.

Der **Franzis-Verlag** hat seit Jahrzehnten qualitativ hochwertige Elektronik-Bücher in seinem Programm. Wir wollen unseren Erfolg auch im Bereich der Computer-Literatur ausbauen.

Deshalb suchen wir kompetente

BUCHAUTOREN

mit guten Erfahrungen auf folgenden Gebieten:

- **Betriebssysteme**
- **Programmiersprachen**
- **Anwenderprogramme**
- **Hardware**

besonders für den Bereich der PC- und PS/2-Systeme.

Sie sollten mit uns zusammen ein attraktives Computer-Buchprogramm mit hohem Qualitätsniveau und hoher Aktualität aufbauen.

Wenn Sie bereits Erfahrung als Autor haben und auch Spaß daran haben, Ihr Wissen an andere in schriftlicher Form weiterzugeben, sollten Sie nicht zögern, mit uns Kontakt aufzunehmen.

Die Voraussetzung bei unserem gemeinsamen Vorhaben ist natürlich, daß Sie sich auf einem der obengenannten Gebiete fundiert auskennen.

Nehmen Sie Kontakt mit uns auf:

M. Heysinger, Chef-Lektor

Franzis-Verlag GmbH, Karlstraße 37, 8000 München 2, Telefon 0 89/51 17-2 42

Philipps-Universität Marburg

Im **Fachbereich Physik** sind zum nächstmöglichen Zeitpunkt zwei Stellen für

Elektroniker/in

zu besetzen. Vergütung erfolgt nach BAT.

Aufgabengebiet: Entwicklung, Aufbau, Inbetriebnahme und Wartung von teils rechnergesteuerten elektronischen Meß- und Datenverarbeitungsgeräten.

Voraussetzungen: abgeschlossene Ausbildung als Techniker/in oder Meister/in oder vergleichbare Qualifikation. Praktische Erfahrung mit Kleinrechnern und Mikroprozessoren (Hard- und Software).

Bewerbungen mit den üblichen Unterlagen sind bis zum **26. 3. 1988** an den
Dekan des Fachbereichs Physik der Philipps-Universität Marburg, Renthof 5, 3550 Marburg
zu senden.

Qualifizierte Frauen sind ausdrücklich zur Bewerbung aufgefordert und werden bei gleichwertiger Eignung bevorzugt eingestellt.

Schwerbehinderte Bewerber(innen) werden bevorzugt.

INSERENTENVERZEICHNIS

Abor 30, 31
ACCOM 134
AD Computertechnik 142
Ahlers 137
Almik 136
AMI 144
A.S.S.-Ware 134

Bajic 135
Baron 33
Binder 39
Bockstaller 136
bsb Datentechnik 137

Chicony 145
Christiani 134
Commodore 121
Computer Discount 178
Computerring Kolter 134
Conex 30, 31
Confident 162

DaCom 29
Dalvo 141
Data Becker 10, 11
Datatech 27
Dawicontrol 173
De Ueno 159
Distec 23
Disys 178
Dobbertin 136
DSM 13
DVS 177

Ecosoft 139
ees GmbH 138
Elco 37
Elektronikladen 34, 35
Elitegroup 146
Emis 138
Ensontech 162

Fölsche 133
Forth 134
Franzis 157, 170, 171, 176
Fujitsu 7

Geosoft 103
Graf 21
GWK 134

Haase & Menrad 135
Hardline 147
Holco 159
Hollmann 172

Imunelec 172
Indutronic 3
Infinity 148
Int. Meßsysteme 136
Intec 135
Interest-Verlag 18
Interplanetary 149
Intra 150

Jeschke 135
Joint Precision 160

King Phoenix 160
Koch CSM 168
Koga 163
Krickl 137
Krischer 175
Kwem 9

Laytronic 138
Leadman 161
Lead Year 151
Linden, v. d. 139

Mannesmann Tally 15
Mathes 167
McGraw-Hill 169
MCI 92, 93, 183
Memory Electronics 133
Meyer, E. W. 30, 31
Micom 135
Microcomp 158
Micromint 164
Miele Datentechnik 138
Müller, Dr. Gert 137
Monterey 152
MoVe 137
MWB 177

NEC 24, 25
Niedermeier 138

Osborne 109

Pandasoft 177
Plantron 2

Röntgen Software 134
Radio-RIM 179
Ranfft 138
Ratev 135
Rheintec 168
Rose 136
RWL 140

S.C. Special 139
Sanwel 153
Schukat 177
SE Spezial Elektronik 17
Shamrock Software 133
Shamrock Vertrieb 139
Siebert 134
Sound Data 154
Sparkasse 61
STAKOM 169
Suchy 137

Team 155
Tetzlaff 136
Toshiba 75
TSS-Schmitz 136

Ueding 178
Ultima 156

Vanko 161
Vobis 184
Vogel-Verlag 143

W&O Computertechnik 133
Weber 139
Wege 133
Wegner 133
Weiss 137
Welter 178
Wienecke 134
Wiesemann & Theis 164
Wilhelm 138
Wilke 133

Z + M 175
Zecks 136

Die Mikrocomputer-Zeitschrift

Verlag: Franzis Verlag GmbH, Karlstraße 37–41, 8000 München 2
Postfach 37 01 20, 8000 München 37
Telefon (0 89) 51 17-1
Telex 5 22 301
Telefax (0 89) 51 17-3 79

Geschäftsführer:
Peter G. E. Mayer, Michael-Alexander Mayer

Verlagsleitung: Peter Habersetzer, Eugen Wintersberger

Anzeigenleitung: Dietger Kötter

Anzeigenverkaufsleitung: Hans-Joachim Hecht (-3 86)

Disposition: Ingrid Daschner (-2 97)

Anzeigenvertretung Inland:

Norddeutschland: Lita Lange, impulse medien service GmbH, Holtenklinker Str. 9, 2050 Hamburg 80, Tel. 0 40/ 7 24 20 37/38
Hessen: Günter Junne, Victor-Achard-Str. 30, 6380 Bad Homburg v. d. H., Tel. 0 61 72/3 33 94
Bayern: Elfie Rusch, Münchner Verlagsvertretung, Sperberstr. 23, 8000 München 82, Tel. 0 89/4 30 73 32
Nordrhein-Westfalen: Bandelow + Partner GmbH & Co. KG, Herr Werner Bandelow, Dellestr. 36, 4000 Düsseldorf 12, Tel. 02 11/20 14 64
Baden Württemberg: Ulrich G. Felger, Honoldweg 27, 7000 Stuttgart 1, Tel. 07 11/63 27 18
Berlin: Rainer W. Stengel, Bischofsgrüner Weg 91, 1000 Berlin 46, Tel. 0 30/7 74 45 16

Anzeigenvertretung Ausland:

Belgien: ECI/United Media International S.A., Avenue de la folle chanson, 2 bte 7, 1050 Bruxelles, Tel. 02/6 47 31 90, Telex 63 950 eci um
Großbritannien: Martin Geerke, Friary Hall, Flat 3, Firary Road, South Ascot Berks. SL5 9HD, Tel.: 9 90/2 86 49 oder 2 76/6 28 28, Telex 8 58 328
Japan: International Media Rep. Ltd., 2–29, Toranomon 1-chome, Minato-ku, Tokyo 105, Tel. 5 02-06 56, Telex 22 761
USA: DemoNet Inc., 7310 Adams Street, Paramount, CA 90723, Tel. (2 13) 4 08-19 68
Frankreich: Agence Gustav Elm, 41, Avenue Montaigne, 75008 Paris, Tel. 01-47 23 32 67, Telex 2 90 260
Italien: Rancati advertising, Milano San Felice Torre 5, 20090 Segrate, Tel. 02-7 53 14 45, Telex 3 11 250 PP MI I
Schweiz: Exportwerbung AG, Kirchgasse 50, 8024 Zürich, Tel. 01-47 46 90, Telex 8 12 765

Anzeigen nach Preisliste Nr. 8, gültig ab 1. 10. 1987

Vertriebsleitung: Peter Habersetzer

Abonnement: Christa Fischer

Handelsverkauf: Dietlind Stauder

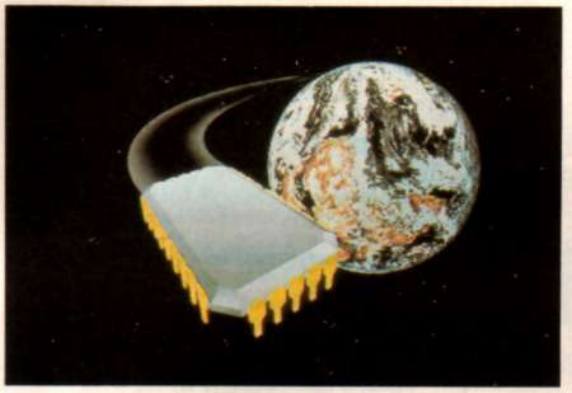
Erscheinungsweise: Die mc erscheint monatlich, jeweils montags am Monatsanfang bzw. am Ende des Vormonats; im 8. Jahrgang

Bezugspreise Inland: Einzelheft 7.– DM, Jahresabonnement 70.– DM; Vierteljahresabonnement 18.50 DM. Studenten, Auszubildende und Rentner erhalten das Jahresabonnement gegen Nachweis günstiger. Der Versand ist im Abonnementspreis eingeschlossen. In den Preisen ist die gesetzliche Mehrwertsteuer in Höhe von 7 % enthalten.

Verlagsvertretungen Ausland:

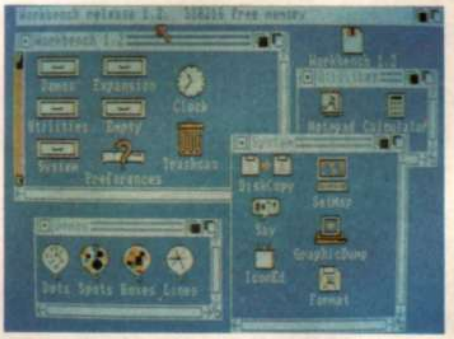
Belgien: Office International des Périodiques (O.I.P.), Avenue Marnix 30, B-1050 Brüssel – Einzelheft 182.– bfr, Jahresabonnement 1963.– bfr.
Dänemark: Harck + Gjellerups Booksellers Ltd., Fiolstraede 31–33, DK-1171 Kopenhagen K. – Jahresabonnement 316.– dkr.
Frankreich: Librairie Parisienne de la Radio, 43, rue de Dunkerque, F-75010 Paris.
Luxemburg: Messageries Paul Kraus, 5, rue de Hollerich, L-Luxembourg.
Niederlande: De Muiderkring BV, Electronics House, Postbus 313, 1380-AH Weesp – Einzelheft 8.85 hfl, Jahresabonnement 89.50 hfl.
Österreich: Erb-Verlag Ges.m.b.H. & Co. KG, Buch- und Zeitschriften-Vertrieb, Amerlingstr. 1, A-1061 Wien. – Einzelheft 60.– öS, Jahresabonnement 620.– öS.
Schweiz: Verlag Thali AG, CH-6285 Hitzkirch/Luzern – Einzelheft 7.– sfr, Jahresabonnement 67.20 sfr (je nach Kurs).

VORSCHAU



Der Prozessor 68030

Kaum hat der 68030 die Erprobungsphase hinter sich, zeigt mc die Vorteile dieses Prozessors und erläutert die dahinter stehende Philosophie, die sich kurz gefaßt etwa so ausdrücken läßt: Voll kompatibel zum Vorgänger 68020, aber ganz, ganz anders ... Ein Insider hilft, den Prozessor mit Harvard-Architektur und MMU richtig einzuordnen!

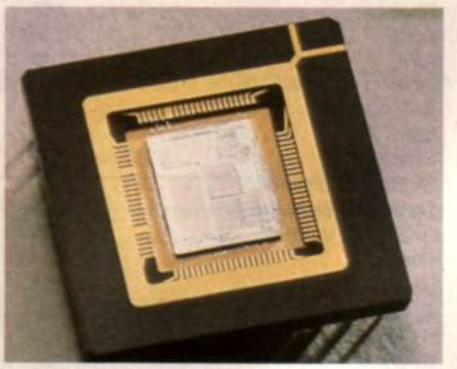


Amiga mit Intuition

Gleich von zwei Seiten her geht mc die Programmierung der grafischen Bedienoberfläche des Amiga an: Sowohl von C als auch von Assembler aus beschreibt die nächste mc-Ausgabe, wie man Intuition in den Griff bekommt. Öffnen und Schließen von Libraries, Screens und Bildschirmfenstern – eigentlich gar kein Problem, wenn man weiß, wie es geht. Und eben das steht (wie immer mit Beispielprogrammen) in mc.

Digitale Bildverarbeitung

Von der Quantifizierung bildhafter Daten über die Bearbeitung der gespeicherten Informationen durch verschiedene Algorithmen bis zur Visualisierung der Ergebnisse auf Monitoren, Plottern oder Druckern werden die verschiedensten Verfahren zur Digitalisierung von Bildern vorgestellt.



mc-Transputer-EMUF

Größte Leistung auf kleinstem Raum zum günstigsten Preis – das sind die herausragenden Merkmale unseres Transputer-EMUFs. Dieser Einplatinencomputer macht modernste Technik leicht anwendbar. Beliebig viele Transputer-EMUFs können über die Links parallel arbeiten. Mit mc gelingt der Einstieg in die Transputertechnik.

Laptop mit Plasmabildschirm

Tragbare MS-DOS-Rechner werden immer beliebter. Trotz der kleinen Abmessungen von Laptops braucht man bei den hochwertigen Systemen auf nichts mehr zu verzichten. Ein hochauflösender Bildschirm und eine Festplatte sind bei ihnen schon Standard. Als einen Vertreter der Laptop-Spitzenklasse haben wir den Rein-300SLC, der mit einem Plasmabildschirm und einer 20 MByte Festplatte ausgestattet ist, auf dem Prüfstand auf Herz und Nieren geprüft.

Aktuelle Sammeldisketten zu mc

MS-DOS-Sammeldisk 12
Diskette zum „PC-Sonderheft" (u. a. CLK. ASM, BAUM.ASM, TERM.BAS, DRUCKER. ASH, Hercules-Grafik, PC-Datentransfer, EGA-Grafik, Gameport, Disk-Speedy, Disk-Reparatur, Musik nebenbei). Preis: 25 DM.

MS-DOS-Sammeldisk 13
Diskette zum mc-Sonderheft „Turbo-Pascal im Einsatz" mit Pascal Quellcode.
Preis: 25 DM.

MS-DOS-Sammeldisk 14
Cross-Assembler und EPROM, Move-Utility für MOS-DOS und Zeitanzeige in Fenstertechnik aus mc 5/87; Wahrheitstafel für Logik-PROMs und schneller Draw-Line-Algorithmus aus mc 6/87; Hardcopy von CGA-Farbgrafiken und Flächen-Füll-Algorithmus aus mc 7/87; vier Bildschirmseiten in Turbo-Pascal, MS-DOS mit Komfort und Extended Memory (> 1 MByte) beim IBM-AT aus mc 8/87. Preis: 25 DM.

MS-DOS-Sammeldisk 15
256K-EGA mit variabler Bildgröße, Hercules-Grafiken drucken, Physik und Gehirn (KI) und Backup ganz bequem aus mc 9/87, Primzahlen-Sucher, pseudozufällige Binärfolgen, Utility für die Hercules-Karte und Leiterbahnen entflechten (Platinenlayout mit dem Lee-Algorithmus) aus mc 10/87.
Preis: 25 DM.

MS-DOS-Sammeldisk 16
Differentieller Dateivergleich, dynamische Bildausgabe, PC minimiert Schaltfunktionen und Turbo-Residenz (aus mc 11/87). MS-DOS-Filemanager und Treiber-Implementation in MS-DOS (aus mc 12/87). Außerdem: Jahresinhalt mc 1987. Diese Diskette wird auch über einen verbesserten Editor verfügen mit einer gewissen Toleranz gegenüber falschen Suchbegriffen und mit einfacher, menügeführter Bedienung.
Preis: 25 DM.

MS-DOS-Sammeldisk 19
Turbo-Pascal ohne Absturz, Linien in C (aus mc 1/88); GALs programmieren (aus mc 1–2/1988); Festplatte sicherer gemacht (aus mc 2/88).

Für den ATARI-ST gibt es drei Disketten:

ST-Sammeldisk 1
EPROM-Programmer und Taschenrechner aus mc 9/87 (TOS-Version vom 27. 11. 86 oder später nötig). Preis: 25 DM.

ST-Sammeldisk 2
Grafikdemo und Minimon aus mc 5/87.
Preis: 25 DM.

ST-Sammeldisk 3
3-D-Darstellung von Funktionen in GFA-Basic aus mc 7/87, Assembler-Routinen in GFA-Basic und Programme zum Setzen des Uhrenbausteins DS1216E aus mc 9/87.
Preis: 25 DM.

Fragen Sie nach den neuesten Disketten.
Bestellung ☎ 0 89/59 54 68

Ausgabe 4/88 erscheint am 28. März 1988

MCI

PERSONAL COMPUTER

Bitte neueste Prospekte anfordern!

Auf alle Geräte 12 Monate Garantie. Preise gültig ab 1.12.87. Lieferbedingungen auf Anfrage. MCI MICRO COMPUTER INSTRUMENTS GMBH eingetragen AG Bergisch Gladbach HRB 2575 · Herstellung und Vertrieb von Microcomputern.

MCI

Bensberger Straße 252 · 5060 Bergisch Gladbach 2
Tel. (02202) 1080
Fax: (02202) 31009 · Telex: 8873518

Schnell, schneller, am schnellsten:

Die Lingo® IBM-AT-Kompatiblen

Es gibt IBM-PC-Kompatible mit dem INTEL 8088 Microprozessor. Die sind gar nicht schlecht. Aber IBM-AT-Kompatible sind besser: Weil der eingebaute INTEL 80286 schneller ist. Aber am schnellsten sind IBM-AT-Kompatible mit dem INTEL 80386. Das ist ein echter 32 Bit-Prozessor. Unser neuer Lingo®386 TOWER AT (rechts) schafft damit im Landmark-Test 22 Mhz.

Lingo 286 MINI (unten)

INTEL 80286 Prozessor, 640 K Speicher 1.2 MB Diskettenlaufwerk, serielle, parallele Schnittstelle, Clock/Calender, Herkules-kompatible Monografik-Karte, große deutsche Tastatur mit „Doppel-Klick" (am besten ausprobieren), MS DOS 3.2 + GW BASIC

VOBIS-Preis mit 20 MB-Festplatte (ohne Monitor) **2998.-**

mit 40 MB-Festplatte (21 msek) (ohne Monitor) **3998.-**

Lingo 386 TOWER(rechts)

INTEL 80386 Prozessor (32 Bit), umschaltbar 17/22 Mhz (nach Landmark-Test), 2 MB Speicher, 32 Bit Slot für z.B. Speichererweiterungskarte, 5 Slots 16 Bit, 2 Slots 8 Bit, 1.2 MB Diskettenlaufwerk, 40 MB-Festplatte (21 msek), 3 weitere Laufwerkeinschübe. Serielle, parallele Schnittstelle, Herkules kompatible Monografik-Karte, große deutsche Tastatur mit „Doppel-Klick" und MS DOS 3.2 und GW BASIC.

mit 40 MB-Platte **8444.-** ohne Monitor

NEU!
HIGHSCREEN EGA-Monitor
25 Mhz, Auflösung max. 720 x 350
798.-

PRISMA EGAMAX 860-Karte dazu
800 x 600 Punkte, auch für Multisynch-Monitor
398.-

Lingo 386 TOWER
Komplett mit EGAMAX 860-Karte und HIGHSCREEN EGA-Farbmonitor
statt (Summe der Einzelpreise) ~~9641.-~~
Komplett nur 9444.-

Lingo 286 MINI (20 MB-Platte)
Komplett mit EGAMAX 860-Karte und HIGHSCREEN EGA-Farbmonitor
statt (Summe der Einzelpreise) ~~4194.-~~
Komplett nur 3998.-
4998.- mit 40 MB-Platte

NEUERÖFFNUNG ab 30. Januar'88 auch in DUISBURG 1 Fr.-Wilhelm-Str. 30

NEU! Lingo Net II
Profi-Netzwerk. Starter-Kit mit 2 Netzwerkkarten Kabel + Software: (u.a. NOVELL-Emulator) **1298.-**
jede weitere Zusatzkarte **565.-**

kompetent + preiswert

WICHTIG! Herstellerbedingte Lieferzeiten! Aufgrund erhöhter Nachfrage ist nicht immer alles sofort lieferbar!

VOBIS
Deutschlands umsatzgrößter Microcomputer-Spezialist

VERSAND-ZENTRALE:
Postfach 1778
Rotter Bruch 32-34
5100 AACHEN
☎ 0241/500081
Tx 832389vobis d

FILIALEN:

1000 BERLIN 30
Kurfürstenstr. 101 · 030/2 13 94 80

2000 HAMBURG
Krohnskamp 15 · 040/2 79 46 76

2300 KIEL
Sophienblatt 74-78 · 0431/67 86 22

2800 BREMEN
Violenstraße 37 · 0421/32 04 20

3000 HANNOVER
Berliner Allee 47 · 0511/81 65 71

4000 DÜSSELDORF
Wielandstr. 21 · 0211/35 99 64

4150 KREFELD
Ostwall 92 · 02151/80 07 93

4300 ESSEN
Huyssenallee 3 · 0201/23 17 74

4600 DORTMUND
Hamburger Str. 110 · 0231/57 30 72

4800 BIELEFELD
Herforder Str. 106 · 0521/6 38 78

5000 KÖLN
Mathiasstr. 24-26 · 0221/24 86 42

2x 5100 AACHEN
Viktoriastr. 74 · 0241/54 31 00
Großkölnstr. 60 · 0241/2 44 94
(gemeinsam mit Foto PORST)

6000 FRANKFURT
Frankenallee 207/209 · 069/73 40 49

7000 STUTTGART
Marienstr. 11-13 · 0711/60 63 36

7500 KARLSRUHE
Kriegsstr. 27/29 (am BGH) 0721/37 82 68

7750 KONSTANZ
Kreuzlinger Str. 18 · 07531/1 55 60

8000 MÜNCHEN
Aberlestr. 3 · 089/77 21 10

8500 NÜRNBERG
Vordere Ledergasse 8 · 0911/23 29 95

8900 AUGSBURG
Jakoberstr. 16 · 0821/152349